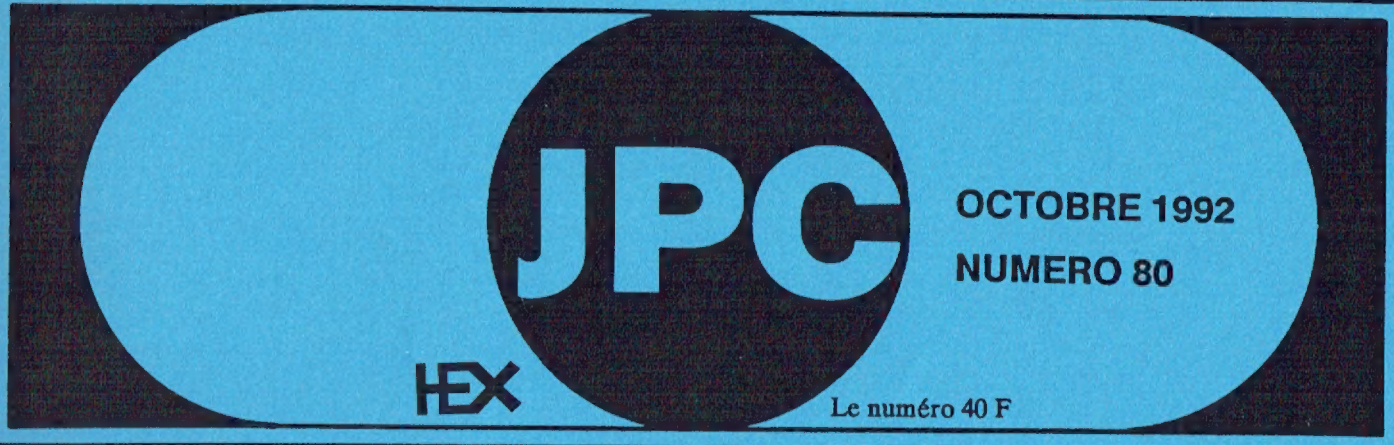


OCTOBRE 1992
NUMERO 80
Le numéro 40 F

A PROPOS DU CLUB

Le Bureau	Editorial	*1*
J. Belin	Compte-rendu de la conférence de Londres	*2*
J. Belin	HEX : un nouveau lien entre les Clubs	*4*
F. Wales / C. Finseth / W M-J...	Calculator Hall of fame	*5*
	Courrier du coeur	*7*

HP-48

M. Shyam	Nouvelles cartes Sparcom	*10*
G. Toublanc	Initiation au RPL	*11*
B. Walsh	Système d'aide	*14*
J. Schwartz	Nouvel index pour les manuels	*19*
C. Vaillant	Initiation au langage machine	*20*
J. Bos	Graphisme et récursivité	*24*

HP-95

R. Young	Pager et mobidem	*28*
L. Thomas	Interface HP-IL	*30*
J. Belin	Nouveaux produits	*31*

EDITORIAL

L'événement de ce mois de septembre, pour les clubs d'utilisateurs de calculateurs HP, fût sans aucun doute la conférence qui à eu lieu à Londres, les 19 et 20 Septembre. Basée à l'Impérial Collège, une université située en plein coeur de la ville près de Hyde Park, cette réunion avait pour but de fêter les dix ans du club HPCC. Elle s'est passée dans d'excellentes conditions matérielles et dans une très bonne ambiance. Nous tenons à remercier toutes les personnes qui l'ont organisée, en particulier Wlodek Mier-Jedrzejowicz, le président de HPCC, pour son travail et la patience qu'il a montré pendant ces deux jours.

Ce numéro de *JPC* est consacré à cette conférence. En effet, la majeure partie des textes que vous y trouverez sont tirés des présentations qui y ont été faites, excepté les articles de Christophe Vaillant et Guy Toublanc. Les lecteurs anglophones pouront aussi lire au cours de nos réunions mensuelles les *proceedings* ayant trait aux sujets que nous n'avons pas traduits. Sachez aussi qu'il existe une disquette contenant les sources des programmes décrits au cours de ces présentations, ainsi que par exemple une impressionnante base de données (sous forme de fichiers textes) contenant les caractéristiques détaillées de tous les calculateurs développés par HP.

Connaissez vous la loi de Parkinson ? *Plus on dispose de temps pour accomplir un travail, plus ce travail prend du temps*. Nous pourrions dire pour ce journal : *moins on dispose de temps pour faire un JPC, plus vite il est mis en pages*. En effet, ce numéro a été presque totalement rédigé, traduit et mis en pages dans la semaine qui a suivi la conférence, afin qu'il soit prêt pour la réunion du 3 Octobre ! Excusez nous donc s'il est légèrement plus fin que les précédentes éditions. Nous espérons que le nombre et la qualité des informations que vous y trouverez vous fera oublier ce petit détail !

Le bureau

ONCE UPON A TIME IN LONDON...

Une petite centaine de personnes s'étaient déplacées à Londres, pour fêter les dix ans de HPCC. Parmi les personalités de notre petit monde, nous avons pu rencontrer les présidents et membres des bureaux des principaux clubs européens ou américains CCD (Allemagne), STaK (Finlande), PCX (Belgique), PROMPT-HPGC (Pays Bas), STORC (Pays bas), HPX (USA), 48SXTANT (Pays Bas)... PPC Paris était représenté par Asdin Aoufi et moi-même. D'autres personnalités du monde HP étaient présentes : Jake Schwartz (USA) qui a filmé en vidéo la quasi-totalité de la conférence et bien sûr Richard Nelson, le créateur du premier club PPC. Du coté des représentants de HP Corvallis nous avons pu voir Jim Donnelly, qui travaille sur la HP48 et a écrit différents livres sur cette machine, Raan Young qui a participé au développement du HP-75 ou Mark Ellis, le responsable des ventes des calculateurs chez HP Grande Bretagne. Que les personnes non présentes dans la liste m'excusent, mais la place manque pour donner tous les noms !

Ces deux journées étaient constituées de mini-conférences d'une vingtaine de minutes traitant de domaines d'interêt général, de présentation de matériels et de "tutorials" sur certains domaines de la programmation sur les HP-48 ou HP95, entre lesquelles nous avions le loisir de discuter entre nous ou de regarder ce que nous proposaient les stands de quelques distributeurs anglais de matériels HP, en ayant dans les mains (en plus de notre machine préférée) une boisson que les organisateurs avaient pris soin de garder au frais pour l'occasion...

En résumé, voici ce qui s'est passé...

Les mini-conférences d'interêt général

Plusieurs conférences on traité du passé, du présent et du futur. James Donnelly a parlé de la période allant de l'antiquité avec les abbaques jusqu'à la HP-35, en passant par la règle à calcul et la Babbage machine. Ensuite, Craig Finseth a établi le lien entre cette machine et le présent, puis David Frodsham nous a donné son opinion sur ce qui pourrait être le futur.

Mark Ellis nous a donné la vision de HP, concernant les palmtops et les raisons qui ont abouti à la création du HP95, à savoir le besoin d'avoir une machine de moins de 300 grammes, avec un accès aux principales fontions par l'appui d'une seule touche.

Le dimanche, d'autres conférences ont traité de l'utilisation des calculateurs dans le monde étudiant, coférences faites par Wlodek Mier-Jedrzejowicz et Lee Woodriff.

Les présentations et annonces de matériels

Ces deux jours de réunion ont bien sûr permis à quelques constructeurs et distributeurs de nous présenter les derniers accessoires et logiciels disponibles sur le marché.

Si les membres de HP ne s'étaient pas déplacés avec une toute nouvelle machine, Raan Young, de HP Corvallis, nous a cependant présenté de façon complète le tout nouveau système de messagerie radio adapté pour le HP95, système développé en coopération avec Motorola. Vous trouverez le texte de cette présentation dans la rubrique HP95.

Une autre société, Sparcom était représentée par son président, Megha Shyam. Celui-ci nous a présenté ses nouvelles cartes destinées aux HP-48 et HP95 (voir dans les sections destinées à chaque machine).

Colin Mason, représentant Mitsubishi, nous a présenté l'état de l'art en ce qui concerne les cartes mémoires. En résumé, les efforts de la société portent sur deux domaines : la diminution de l'épaisseur des puces (pour obtenir des cartes les plus fines possibles) et l'augmentation de leur capacité. Dans un très proche avenir, il sera possible d'obtenir des cartes de mémoire dynamique (DRAM) de 24 Mo. Si ce cartes ne sont pas prévues pour être installées sur le HP95, il pourrait en être autrement pour les "Mixed Cards", comportant à la fois de la mémoire FLASH et de la mémoire Statique (SRAM), dans des proportions allant jusqu'à 1Mo de FLASH et 1Mo de SRAM.

Enfin, Richard Nelson, troquant son rôle d'ancien président du club PPC USA pour celui de commercial de la société EduCALC, dont il fait partie depuis quelques années, nous a présenté quelques nouveaux produits fabriqués par des sociétés indépendantes. Le plus interessant étant sûrement le SunDisk, une carte FLASH au format PCMCIA, adaptée pour le HP95, dont les capacités vont jusqu'à 10 Mo. Ce produit ne sera annoncé officiellement que le 20 Octobre !

Les Tutorials

Quelques sessions de la conférence ont été consacrées aux réalisations et à l'explication de quelques points concernant la programmation sur les calculateurs HP :

En ce qui concerne le HP-48, Brian Walsh, le président du Club Américain HPX, nous a présenté son programme permettant d'obtenir un aide en ligne sur les fonction de cette machine. Ensuite, ce même Brian Walsh accompagné cette fois-ci de Wlodek Mier-Jedrzejowicz, puis James Donnelly, nous ont présentés les nouveaux outils (venant en plus de ceux que nous connaissons déjà) permettant de développer des applications performantes sur cette machine, en System RPL. Jurgen N. Bos nous a présenté quant à lui une utilisation originale de la fonction PLOT. Enfin, Jake Schwartz nous a présenté son imposant travail consistant à faire un meilleur index pour les manuels de la HP-48.

En ce qui concerne le HP95, Craig Finseth (qui a travaillé sur le développement du HP-94) nous a présenté quelque chose de totalement inédit (ou interdit) pour les personnes non-acréditées par HP, à savoir le développement des applications *System Manager* sur le HP95 ! Ceci sous le regard bienveillant des autres représentants de la société... Peu après Graeme Casey a un peu plus enfoncé le clou en nous parlant de façon détaillée du développemnt des application tenant compte des possibilités multi-langages de la machine.

Illogical Mystery Tour

Un des grands moments de cette réunion fût la Présentation du "Corvallis Plant Tour", une vidéo réalisée par Raan Young et Frank Wales. Sur un ton totalement délirant, qui montre que l'ambiance dans les laboratoires de Corvallis semble, disons... très détendue, nous découvrons le fameux "Hall Of Fame" constitué de tableaux montrant la quasi totalité des calculateurs fabriquées par HP. Ensuite, nous avons pu découvrir que l'assemblage du HP95 est (mis à part pour la carte mère) totalement effectué manuellement !

Peu après, une séance de questions-réponses a eu lieu avec les représentants de HP, mais ceux-ci ayant annoncé dès le début qu'ils ne répondraient pas aux questions concernant les projets en cours, les questions n'ont pas dépassé les interrogations d'ordre général. Une des seules informations a avoir retenu notre attention est que, à Corvallis, le processeur Saturn est quasiment arrivé en fin de son cycle de vie. Ce qui veut dire que si la HP48 a encore quelques années devant elle, il est peu probable que son successeur utilise ce circuit. Un petit point intéressant est que cette séance s'est déroulée sous l'affichage d'un transparent montrant un Jaguar, avec la légende "Considering the Cat", officialisant le fait que les successeurs du HP95 porteraient des noms de félins. On peut regretter que les gens de Corvallis n'aient pas pensé nous faire passer un message, en nous présentant, par exemple un Cougar... A moins bien sûr que le félin que nous avions sous les yeux soit plutôt une panthère, qui (vérification faite après mon retour, alors que je n'avais plus la photo sous les yeux) possède quasiment le même pelage si ce n'est une petite différence...

L'Anniversaire du HP-35

Le samedi, en fin d'après midi, une petite fête été organisée afin de célébrer les vingt ans de la HP-35, la première calculatrice scientifique. A cette occasion, nous avons pu déguster un magnifique gateau représentant cette machine (avec touches en vraies couleurs et en relief !). Ensuite, un tirage au sort a désigné les heureux gagnants de nombreux lots dont des calculateurs HP, un HP-71 et un gros lot constitué d'un HP95, gagné par le président organisateur Wlodek Mier-Jedrzejowicz (et ce n'était pas truqué !). Pour notre part nous n'avons malheureusement rien eu à déclarer à la douane à notre retour !

La vente aux enchères

Les heureux gagnants du tirage au sort précédement cité, qui ne trouvaient pas leur lot à leur goût ont peut être profité de l'évènement suivant pour se débarrasser de cet objet pourtant pas très encombrant. En effet, en fin d'après midi, une petite vente aux enchères a été organisée, dans le but de vendre une centaine de machines et de périphériques d'occasion. En général, la plupart des objets (y compris un HP-71) sont partis entre 15 et 40 £. Vingtième aniversaire oblige, c'est une HP-35 qui est montée le plus haut, avec une adjudication à 190 £.

La réunion des présidents des Clubs

Le samedi soir, après un bon repas, les présidents des clubs se sont retrouvés afin de discuter de leur situation et des moyens de rendre les échanges d'informations et de contacts plus réguliers. Cette discussion a abouti à la création de HEX, dont vous trouverez plus d'informations dans un des articles suivants.

La Visite de la "Babbage Machine"

En clôture de ces deux jours fertiles en surprises, nous avons eu le privilège d'assister, dans le Science Museum (l'équivalent de notre Palais de la Découverte), à une démonstration privée du fonctionnement de la *Babbage Machine*. Considéré comme étant le véritable ancêtre de l'ordinateur, cet imposant ensemble de leviers et d'engrenages de plus de 3.3 m de long et plus de 3 tonnes a été conçu par Charles Babbage en 1847. Capable de calculer des

polynômes du 7eme degré sur 30 digits, cette machine utilise la méthode dite *des différences finies* pour calculer les différentes tables. N'ayant jamais pu être terminée a l'époque, celle que nous avons vu au musée est une reconstitution datant de 1991, date du bicentenaire de la naissance de Charles Babbage.

Regrets éternels

Malgré cet excellent week-end, il nous reste quelques regrets. Premièrement, que HP n'ait pas profité de la fête pour nous parler de ses projets immédiats en ce qui concerne les prochaines machines. Ensuite de ne pas avoir pu assister à la démonstration du train électrique contrôlé par un HP-48, son créateur, Robert Pulluard ayant dû écourter son séjour à Londres à cause de problèmes de santé. Enfin, que nous n'ayons pas pu assister au Dîner de Clôture, célébrant les dix ans de HPCC. Ceci à cause de notre obligation de retourner en France le soir même, avec des horaires de transports inadaptés à l'occasion.

Quoi qu'il en soit, ceci n'a pas assombri un week-end qui s'est revelé exceptionnel.

Jacques Belin (123)

EUROPE CONNECTION

Un autre grand événement, au cours de ce week-end, fût la réunion des présidents des clubs présents à la conférence. Afin de renforcer les contacts, la coopération entre les différents clubs Européens et leur donner plus de poids au cours de discussions face à HP Europe dans le cas de problèmes communs et coordonner toute action commune, il a été créé une entité, nommée *HEX* (Handeld European eXchange).

HEX obéit aux règles suivantes :

- Il est créé une structure (centre de coordination), simple et aussi économique que possible, collectant toutes les informations provenant des clubs membres de HEX, afin de les redistribuer aux autres clubs.

- Pour des raisons pratiques, une seule langue (l' Anglais) sera utilisée pour les communications au sein de HEX.

- Chaque club délègue un des membres de son bureau, pour tous les contacts avec HEX. Cette personne doit avoir une bonne connaissance de l'Anglais et doit être parfaitement connue (nom, adresse personnelle, téléphone...) par les autres clubs.

- Au moins deux fois par an, chaque club doit envoyer un rapport décrivant son statut actuel au centre de coordinantion. Par "statut actuel" il est entendu toute information concernant les activités et autres projets, ou au contraire toute information concernant un manque d'activité un une suspension de parution des journaux.

- Une liste complète de tous les clubs appartenant à HEX doit être publiée et rendue disponible à l'intérieur de chaque club. Cette liste pourra être utilisée dans toute action de promotion de HEX, au niveau Européen.

- Un Logo commun (voir couverture) pourra être utilisé par chaque club (sur le journal, la correspondance ou sur les lieux de réunion...) sans devoir être plus présent que le titre et le logo de chaque club.

- Tous les clubs acceptent d'échanger leur journal avec les autres clubs membres de HEX. Un *English Summary* devra être ajouté à tous les journaux non écrits en langue Anglaise.

- Les articles parus dans un journal pourront être publiés dans les journaux des autres clubs membres de HEX. L'origine et le nom de l'auteur devront être mentionnés.

- Le premier centre de coordination de HEX sera tenu par le Club Hollandais PROMPT HP-GC.

Tout autre tout club européen, acceptant les conditions précédentes, peut rejoindre HEX à tout moment.

A la date du 19 Septembre 1992, jour de sa création, HEX est composé des clubs suivants :

CCD E.V.
Postfach 11 04 11
6000 FRANKFURT AM MAIN 1
ALLEMAGNE

PCX BRUGE
Postbus 205
8000 BRUGGE 1
BELGIQUE
Tel : +32 50 827 979

STaK
c/o Tapani Tarvainen
Wivi Lönnin Katu 7A 19
40100 JYVASKYLA
FINLANDE

HPCC
c/o Wlodek Mier-Jedrzejowicz
42, Heathfield Rd
London W3 8EJ
GRANDE BRETAGNE
Tel : +44 81 992 2039

BITS HP48 Sig
c/o A. R. Duell
H. H. Wills Physic Lab
Tyndall Avenue
BRISTOL BS8 1TL
GRANDE BRETAGNE

PROMPT HP-GC
Gerlof DONGA
Postbus 1081
1500 AB ZAANDAM
PAYS BAS
Tel : +31 75 704 205

STORC
Postbus 205
5600 AP HEINDHOVEN
PAYS BAS
Tel : +31 40 410 049

Et bien sûr PPC Paris, dont vous connaissez déjà l'adresse !

Notons aussi que HEX est un instrument de communication entre les bureaux des clubs. Dans le cas ou un adhérent d'un club souhaite aborder un sujet pouvant intéresser les clubs membres de HEX, il devra s'adresser en priorité à son club, et non au centre de coordination ou à la totalité des clubs membres de HEX.

Une des toutes premières actions de HEX sera d'étudier la possiblité de faire inclure dans chaque boite de calculateur HP vendu en Europe, un carton annonçant l'existence des clubs membres du groupement et mentionnant leurs adresses respectives. Cette action est donc équivalente à ce qui a été fait il y a quelques années (mais seulement au niveau français) par PPC Toulouse et PPC Paris, avec la HP-41, et qui nous avait fait connaitre par un très grand nombre d'utilisateurs.

Jacques Belin (123)

CALCULATOR WALL OF FAME

Dans un hall de l'usine Hewlett-Packard à Corvallis, des panneaux accrochés aux murs montrent des exemplaires de la plupart des calculatrices et accessoires fabriqués par HP. A coté de chaque machine, on peut trouver une étiquette montrant le nom de code utilisé pendant le développement, la date d'introduction et le prix d'introduction (NDLR : non listé dans cette liste). Les machines sont regroupées en tableaux, et chaque groupe possède aussi un code interne. Cet affichage est connu par le personnel sous le nom de "Calculator Hall of Fame". Le tableau suivant montre ces informations telles qu'elles ont été notées par différents visiteurs au cours de la visite de l'usine. Ce type d'information n'est normalement pas rendue officielle par HP, mais cette liste peut être considérée comme "officielle", étant donnés le lieu et la façon dont elle est présentée.

HP-xx	Code	Date	Observations
[CLASSIC]			
35		01/07/72	RPN Scientifique
80		01/02/73	RPN Financière
45	WIZARD	01/05/73	RPN Scient, 10 mem
65	SUPERSTAR	19/01/74	RPN Sci, Prog, cartes
70	SCROOGE	01/08/74	RPN Financière
55	MERLIN	01/01/75	RPN Sci, prog
67	HAWKEYE	01/07/76	RPN SCi, Prog, cartes
[WOODSTOCK]			
21	PUMPKIN	01/02/75	RPN Scientifique
25	SQUASH	01/08/75	RPN Programmable
22	TURNIP	01/09/75	RPN Financière
27	SALAD	01/05/76	RPN Scient/Financ.
25C	SQUISH	01/07/76	RPN Pro mem continue
29C	BONNIE	01/07/77	RPN Sci, prog
[TOPCAT]			
91	FELIX	01/03/76	RPN Sci + imprimante
97	KITTYHAWK	01/07/76	67 avec imprimante
92	BOBCAT	01/07/77	RPN Fin + imprimante
97S	RICOCHET	01/12/77	97 + liaison série
[STING]			
10	KISS	01/07/77	ALG 4 fct + impr.
19C	CLYDE	01/09/77	RPN Sci + impr

[SPICE/SPIKE]

31E	GINGER	01/05/78	RPN scientifique
33E	SAGE	01/05/78	RPN Scient, prog.
38E	CHIVE	01/05/78	RPN Financière
32E	THYME	01/07/78	RPN Scient + Stats
33C	SAGE C	01/07/79	33E + Mem Continue
34C	BASIL	01/07/79	RPN Sci prog SOLVE
37C	PARSLEY	01/07/79	RPN Financière
38C	CHIVE C	01/07/79	38E + Mem Continue

[<NO NAME>]

01	CRICKET	01/07/78	montre calculatrice
01	CRICKET GOLD	01/07/78	la même, boitier or

[COCONUT]

41C	COCONUT	01/07/79	RPN Sci Prog Modules
41CV	SILVERBIRD	15/12/80	41 + 2240 Pas
41CX	HONEYNUT	01/10/83	41CV + Time + XFcts
82140A	HYSTER	01/08/79	lecteur de cartes
82143A	HELIOS	01/08/79	imprimante 24 cols
82120A	HURRICANE	08/04/80	accus rechargeables
82153A	WAND	06/06/80	lecteur codes barres
OPT 001	BLANKNUT	24/03/81	41 sans marquage

[VOYAGER]

11C	1.0	01/09/81	RPN Sci Prog
12C	1.0	01/09/81	RPN Financ. Prog
15C	1.5	01/07/82	RPN Sci SOLVE...
16C	PR	01/07/82	RPN Hexa prog
10C	0.5	02/09/82	RPN Scientifique

[HP-IL]

82160A	GRAPENUTS	14/12/81	module HP-IL HP-41
82161A	FILBERT	14/12/81	lecteur cassettes
82162A	SPECIAL K	14/12/81	imprimante 24 cols
82166A	PILBOX	14/12/81	convertisseur GPIO
82167A	PILCABLE	14/12/81	cable HPIL
82163A	WALLABY	05/04/82	interface video
82165A	FOXNUT	10/05/82	interface GPIO
82169A	NUTSHELL	01/03/83	interface HP-IB
82164A	KUKUINUT	01/05/83	interface RS-232C
82168A	WOMBAT	01/06/83	modem acoustique

[sans nom]

75C	KANGAROO	15/09/82	BASIC HPIL+Cartes
82713A	FALINE	01/03/83	Pug-In Mod. Simul.
71B	TITAN	01/02/84	BASIC HPIL...
75D	MERLIN	01/02/84	75C av codes barres
82718A	POD	01/07/84	Pod d'expansion

[sans nom]

18C	CHAMPION	01/06/86	ALG Financière
82240A	REDEYE	03/11/86	IR printer
28C	PALADIN	05/01/87	RPL Scientifique
19B	TYCOON	04/01/88	ALG Financière 32Ko
28S	ORLANDO	04/01/88	28C + 32Ko

[PIONEER]

17B	TRADER	04/01/88	ALG Financière
27S	MENTOR	04/01/88	ALG Scient/Financ.
22S	PLATO	01/06/88	ALG Scientifique
32S	LEONARDO	01/06/88	RPN Scient. Prog.
42S	DAVINCI	31/10/88	RPN compatible 41
14B	MIDAS	31/10/88	ALG Financière
10B	ERNST	03/01/89	ALG Financière
20S	ERNI	03/01/89	ALG Scientifique
21S	MONTE CARLO	03/01/89	ALG Probab./Stats

[sans nom]

48SX	CHARLEMAGNE	16/03/90	RPL Scient. Prog.
82211A	FLAMBERGE	16/03/90	Equation Library

Pour compléter, voici quelques autres produits qui ne sont pas sur le Hall Of Fame, du moins à notre connaissance. Ceci pourrait être vérifié après un examen attentif de la cassette "Corvallis Plant Tour" visionnée au cours de la conférence.

82182A	PHINEAS	14/12/81?	Module Horloge
17BII	?	01/90	17 + RPN
19BII	?	01/90	19 + RPN
46	?	<=1973	45 avec imprimante
32SII	NARDO ?	01/03/91	vers. dérivée de 32
48S	SHORTY	02/04/91	48 sans ports
81	?	?	80 avec imprimante
83	?	?	?
94	?	05/86	MS-DOS industriel
95C	?	/	proto non commercial.
95LX	JAGUAR	23/04/91	Pocket PC w/Lotus 1-2-3
95LX1MB	JAGUAR	02/03/92	95 avec 1 Mo RAM
82242A	BLINKY	88?	module IR HP-41
82210A	HOOVER	15/10/90?	Emulateur 41 sur 48
82240B	VISINE	?	Impr IR vers B
2225B	?	?	ThinkJet HPIL
9114	?	?	Lect. disk HPIL
9114B	?	?	Lect. Disk Ver. 2

Vous pouvez constater qu'il reste des points d'interrogation. Ne sont pas listés non plus tous les modules HP-41 et HP-71, les adaptateurs secteurs et autres accessoires annexes. La chasse aux renseignements est donc toujours ouverte...

Frank Wales
Craig Finseth
Wlodek Mier-Jedrzejowicz
.
.
.
traduit et complété par Jacques Belin (123)

COURRIER DU COEUR

Claude MARCOIN
158, rue de la CROIX-NIVERT
75015 PARIS
Tel. : (1) 42 50 48 81

Recherche pour HP-71B :

module RAM 64K (Front Port), module RAM 128K (Card reader Port), module EPROM 64K.

Alain FOUQUET
62, rue de l'égalité
64 la flèche
92130 ISSY LES MOULINEAUX

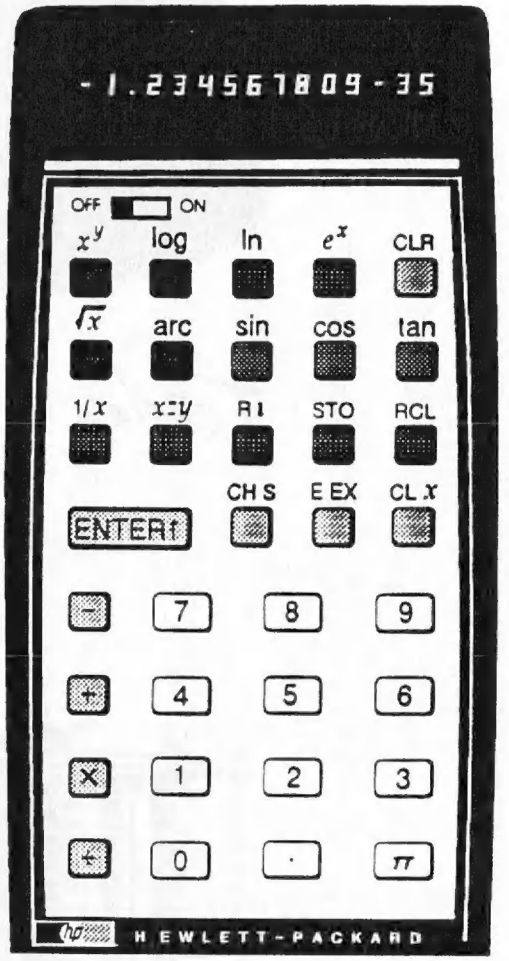
Vend :

1HP41CX, 2 modules XMEMORY, 1 module extended IO, 1 module games, 1 module plotter, 1 module MATH, 1 lecteur de cartes réf 82104A, 1 lecteur de codes barres réf 82153A, 1 imprimante réf 82143A, 1 lecteur de cassettes réf 82161A, 8 cassettes digitales, 2 chargeurs. Toutes les doc et cartes aide mémoire, nombreux softs et bibliothèques. Etat exceptionnel.

Asdin AOUFI
Tél. prof: (1) 49 40 34 34

Vend :

1 ThinkJet pour HP71B et 1 lecteur de cassettes digitales pour HP-41CX ou HP71B. Le tout en parfait état avec accessoires et chargeurs.



Charles Babbage

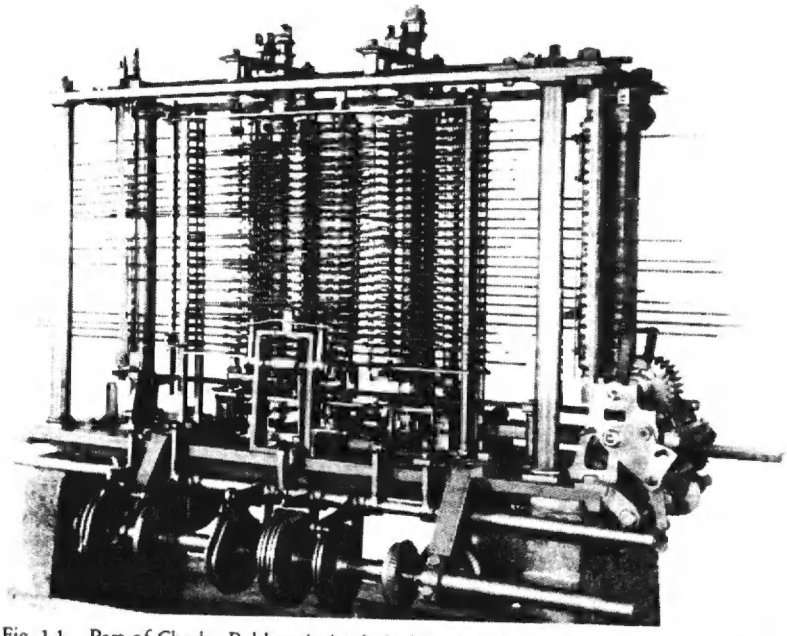


Fig. 1.1 Part of Charles Babbage's Analytical Engine, designed between 1833 and 1837.

HP-48

M. Shyam	Nouvelles cartes Sparcom	*10*
G. Toublanc	Initiation au RPL	*11*
B. Walsh	Système d'aide	*14*
J. Schwartz	Nouvel index pour les manuels	*19*
C. Vaillant	Initiation au langage machine	*20*
J. Bos	Graphisme et récursivité	*24*

LES NOUVELLES CARTES SPARCOM

Depuis la fondation de Sparcom Corp en 1990, nous nous sommes appliqués à mettre en valeur la HP-48SX avec des applications qui permettent à l'étudiant d'apprendre la matière d'un sujet à l'aide d'un environnement portable tel que celui de la HP-48SX. La gamme de produits de Sparcom est conçue pour permettre à l'étudiant d'éxécuter de nombreuses fonctions classiques à l'aide de la HP-48SX.

Deux produits clés ont été introduits cette année, nous avons étendu les capacités de la HP-48SX à l'analyse des circuits linéaires utilisée par de nombreux élèves ingénieurs en électronique. Cette carte s'appelle Spice 48. Permettez-moi de prendre quelques minutes pour mettre en évidence quelques unes des contributions apportées par Sparcom. En développant ce produits nous avons conçu le scénario suivant:

- Dessiner le circuit sur une feuille de papier.
- Créer le schéma du circuit sur la HP48SX.
- Analyser, afficher le résultat.
- Le résultat peut être sauvé ou transmis à un PC.

Spice 48

Le produit supporte les éléments linéaires tels que des résistances, capacités, inductance... L'utilisateur a l'opportunité de réaliser une analyse transitoire ou d'équilibre en courant continu ou alternatif. Les éléments peuvent être identifiés par des labels, les sources de puissance par des labels avec une indication claire de la connection éventuelle du + et - pour chaque élément du circuit. Le circuit conçu peut être transféré à un schéma avec une génération automatique des coordonnées des éléments, de leur orientation et de leur montage.

Différents types d'analyse sont possibles telles que :

- Analyse ponctuelle.
- Analyse de la fonction de transfert.
- Création diagramme de Bode.
- Détermination de la réponse transitoire.

Les résultats du calcul de réponse transitoire peuvent être transmis à travers un gestionnaire de sortie et visualisés pour chaque noeud. Souvent comme les valeurs calculées sont nombreuses nous avons une commande groupée.

Le fichier d'entrée du circuit peut être transferré à un PC. Le prochain objectif pour ce produit est de supporter PSPICE et SPICE afin que l'analyse de sous circuits composés jusqu'à 20 transistors MOS ou bipolaires puisse être supporté par Spice48.

CALCULUS.

Cet été nous avons introduit un produit qui est clairement conçu pour accroitre l'apprentissage du calcul en utilisant des outils de la décennie 90. Notre cible est l'étudiant de première année d'université qui apprend plusieurs nouveaux concepts de calculs pour la première fois. Lors du développement de ce produit nous nous sommes concentrés sur un outil avec des capacités étendues et une bibliothèque de fonctions.

Bibliothèque de fonctions.

En développant cette bibliothèque de fonctions nous nous sommes basés sur des domaines utiles à l'étudiant.

- Ajuster un point et une pente.
- Ajuster deux points et deux pentes.
- Ajuster plusieurs points.
- Définir une fonction par morceaux
- Construire un polynôme à partir de ses racines.
- Rechercher les racines d'un polynôme.
- Calculer un développement de Taylor.

Analyser les possibilités de calcul d'intégrales du calculateur en montrant à l'étudiant la convergence des méthodes d'intégration utilisant les rectangles à gauche, les rectangles à droite, le point milieu, les trapèzes et la règle de Simpson.

Environnement graphique 2D

l'environnement graphique 2D propose une manière puissante d'analyser et de tracer des fonctions, en coordonnées polaires et en représentation paramètrique. La principale caractéristique est le mode TRACE. En mode TRACE, le curseur figure l'équation courante, alors que C et D diminuent et augmentent la variable indépendante, E et B choisissent entre les équations si 'EQ' contient plus d'une équation. 2D et 2C déplace le curseur à la valeur maximale et minimale de la variable indépendante. La courbe est tracée point par point sauf dans les régions où l'équation n'est pas définie. Le mode TRACE affiche les coordonnées pour trois types de fonctions définies ci-dessous:

- **Fonc** - Les coordonnées affichées sont la valeur x du c ur et F(x), la fonction y pour x. Les coordon s sont étiquetées X et Y.

- **Polaire** - Les coordonnées affichées sont la valeur α du curseur et R(α), la fonction de r pour cette valeur. Les étiquettes de coordonnées sont ■ et R.

.Paramétrique - Les coordonnées affichées sont la valeur t de la variable indépendante et F(t), par exemple les valeurs des fonctions x et y pour ce paramètre t. Les étiquettes de coordonnées sont T, X et Y.

Environnement graphique Sparcom 3D

L'environnement graphique 3D fournit une manière puissante pour tracer des fonctions 3D, des courbes paramétriques et des surfaces paramétrèes. Elles sont dessinées av a point de perspective. Une méthode rapide ut la profondeur pour calculer la perspectiv utilisée afin d'éviter de calculer des transform matricielles complexes et coûteuses en tem cessaires pour l'affichage 3D. La résolutio graphisme 3D est controlée finement par le n e de carrés du maillage, de segments de droites ou unité utilisateur.

Tous les graphes 3D sont tracés dans un parallélépipède rectangle, spécifié par les val rs minimales et maximales de x, y, et z. Le vo e affiché est projeté sur le plan de vue avec pour ce e de projection le point de visée. Ce plan de projection est supposé distant d'une unité du point de visée et est toujours parallèle au plan x-z. Le volume par défaut est un cube de 10x10x10 centré en l'origine, avec pour point de visée localisé en [20 -20 20] par défaut. Approcher le point de visée du graphe accroit la distortion de perspective et inversement. En augmentant ou diminuant xoeil et yoeil suffisamment permet de faire tourner le graphe autour de l'axe des z, indépendemment des restrictions sur la position du point de vue par exemple l'oeil.

L'environnement graphique 3D trace automatiquement l'équation spécifiée. La touche du curseur se déplace entre les points affichés du graphe qui peuvent être les intersections des lignes de maillage ou les extrémités des segments. C et D décroissent et accroissent la première variable tandis que E et B acroissent et décroissent la seconde variable sauf pour les courbes paramétrées. Dans tous les cas en préfixant une touche de curseur avec 2 on peut incrémenter ou décrémenter la variable aussi loin que possible dans la direction choisie. Certaines des nouvelles commandes inclues dans l'environnement graphiques sont :

EYE3	Position de l'oeil pour Sparcom 3D, stockée dans la variable réservée PPAR3.
FUNC3	Sélectionnne le type de tracé pour FUNC3.
PCURV3	Sélectionnne le type de tracé pour PCURV3.
PSURF3	Sélectionnne le type de tracé pour PSURF3.
STOX, Y, Z	Stocke x, y, z dans l'équation.
TDOM	Nom de la variable t pour la courbe.
TRES	Résolution pour la variable t.
TVAR	Nom de la variable t.
U,VDOM	Domaines des variables u et v pour la surface.
U,VRES	Résolution pour les variables u et v.
U,VVAR	Nom des variables U et V pour la surface.
VBOX	Dessine une boite dans PICT utilisant les intervalles de variation définis par XRNG3, YRNG3, ZRNG3.

Outil d'analyse vectorielle

Quand les touches du kit d'analyse vectorielle sont installées, nous assignons de nouvelles fonctions aux touches suivantes: -, ., ,, /, 2', 2*, F, 2f, l, 9.
L'utilisateur est alors placé en mode USER. Ces touches prennent de nouvelles significations pour l'analyse de vecteurs réels ou formels. Les nouvelles fonctions sont :

VUNIT	Norme le vecteur.
VCROS	Produit vectoriel; super ensemble de CROSS.
VDOT	Produit scalaire.
VABS	Norme d'un vecteur; super ensemble de ABS.
GRD	Gradient d'un champ scalaire.
DIV	Divergence d'un champ de vecteur.
CURL	Rotationnel d'un champ vectoriel.
LAPL	Laplacien d'un champ scalaire.
V1OP	Opération unaire sur un élément d'un vecteur.
V2OP	Opération binaire sur vecteur.
V+,-,*,/	Addition, soustraction,multiplication,division.
VDER	Dérivée d'un vecteur.
VINT	Primitive d'un vecteur.

Megha Shyam, Sparcom Corp.
Traduit par Asdin AOUFI (562)

DEBUTER EN RPL

Depuis le numéro 78 un certain nombre de fichiers sources de programmes en assembleur sont parus. Dans un premier temps je leur ai donné le même aspect que celui auquel nous étions habitués avec les listings pour HP-71 et HP28 parus dans JPC où dans les livres traitant du langage machine de celles-ci. Cette manière de faire devait permettre d'assurer une continuité pour les habitués utili- sant les compilateurs pour HP-71. Maintenant qu'il est possible de se procurer, auprès du club, les outils de

développement créés par HP il m'a semblé nécessaire de faire des fichiers sources compatibles avec ces outils. A ce propos je signale que ces outils ne sont pas d'une utilisation aussi simple que ceux auxquels nous étions habitués. Il y aura une autre solution : en effet notre ami suisse Laurent Grand a réalisé un compilateur plus simple d'emploi qui compile les trois langages : RPL, Assembleur et User RPL ce que ne fait pas HP. Actuellement il doit faire les dernières retouches car cela est un travail de longue haleine et je pense qu'il pourra bientôt nous le présenter dans sa dernière version.

Pour aujourd'hui mon but est d'aborder simplement la programmation RPL au travers de deux exemples.

Dans un précédent article j'avais vanté les deux avantages de cette programmation : encombrement mémoire réduit et gain de rapidité

Le premier exemple portera sur un programme qui inverse une chaîne. A l'aide d'un petit programme en User RPL :

```
REV.user

«
  DUP TYPE 2 ≠
  IF
  THEN
    514 DOERR
  END
  "" OVER SIZE 1
  FOR j
    OVER j j
    SUB + -1
  STEP SWAP DROP
»
```

nous obtiendrions "LOLULA" → "ALULOL"

Voici une autre solution traitée en RPL avec le programme:

```
REV.rpl
```

Le fichier source à compiler :

```
RPL

 ::                 * début programme
   CK1&Dispatch     * teste s'i y a 1 argument sur
                    * la pile suivant le type pré-
                    * cisé . si erreur émet le mes-
                    * sage correspondant et sortie
   str              * type chaine
                    *      = system binary <3>
   ::               * début programme
   NULL$            * "" départ de la chaine inverse
   OVERLEN$         * OVER duplique la chaine
                    * SIZE = L -> system binary
   #1+              * ajoute 1 à 1 system binary
   ONE_DO (DO)      * 1 L+1 FOR (index non présisé
                    * appelons le j par exemple)
                    * ici nous avons
                    * niveau 2 : la chaine donnée
                    * niveau 1 : la chaine inverse
     OVER           * duplique la chaine donnée
     INDEX@         * (j) index de cette boucle
     DUP            * duplique j
     SUB$           * SUB on extrait le caractère j
     SWAP&$         * SWAP
                    * niveau 2 : caractère j
                    * niveau 1 : chaine inversée
                    * &S ajoute 2 chaines
   LOOP             * NEXT : si j < L+1 la boucle
                    *        continue
   SWAPDROP         * SWAP
                    * niveau 2 : chaine inversée
                    * niveau 1 : chaine donnée
                    * DROP : on se débarasse de la
                    *        chaine donnée
 ;                  * marqueur de fin correspondant
                    * à :: après str
;                   * marqueur de fin pour le pro-
                    * gramme entier
```

Mis sur la pile REV.rpl occupe 42.5 octets contre 91.5 octets pour REV.user.

REV.rpl inverse une chaîne de 208 caractères en 1.6 seconde contre 6.5 secondes pour REV.user.

La prochaine fois vous aurez la solution avec la boucle en assembleur ce qui sera encore plus rapide.

On remarquera que pour la boucle en RPL on fait :

```
1 n FOR ....... NEXT
```

pour obtenir la même chose en System RPL il faut écrire:

```
n #1+ DO ........ LOOP
```

à la limite supérieure de l'index doit être ajouté 1.

Il y a un certains nombre de différences entre les deux langages et auxquelles il faut s'adapter. Ce qui demande une lecture attentive du manuel.

Dans un précédent article je vous avait proposé le programme PIL5 destiné au *coin des codes HP48*. Cette rubrique devant être remaniée ce programme ne se justifie plus sous sa forme originale.

Aussi mon deuxième exemple sera un programme plus utile, car moins spécialisé, et remplaçant PIL5. En effet, si PIL5 traitait une chaîne pour être affichable par lignes de 5 caractères, PIL permet l'affichage par lignes de 1 à 20 caractères.

Ici nous aurons un mixage d'RPL et d'assembleur.

PIL

niveau 2 : la chaîne à traiter soit C$
niveau 1 : 1 réel pour le nombre de caractères
par ligne soit N

Le fichier source à compiler :

```
RPL                     ndique au compilateur
                        le type d'instructions
                        ( RPL )
::                      * début programme
  CK2&Dispatch          * vérifie s'il y a  2 ar-
                        * guments de type à pré-
                        * ciser . si erreur émet
                        * le message correspon-
                        * dant et arrêt
  FORTYNINE             * system binary <49> en
                        * décimal <31> en hexa
                        * niveau 2 : chaîne
                        * donc type 3
                        * niveau 1 : 1 réel
                        * donc type 1
  ::                    * début programme

ASSEMBLE                * indique au compilateur
                        * le type d'instructions
                        * ( Assembleur )
      con(5) #1cc6b     * réel 20

RPL                     * on repasse en RPL

   %MIN                 * minimum de 2 réels
                        * ici MIN( N , 20 ) pour
                        * lignes de 20 caractères
                        * maximum

 ASSEMBLE                * repasse en assembleur

      con(5) #2a2c9     * réel 1

RPL

   %MAX                 * maximum de 2 réels
                        * ici MAXi(arg. ,1) pour
                        * lignes de 1 caractère
                        * minimum donc 0 < N < 21
   COERCE               * conversion
                        * réel -> syst. binary
OVERLEN$                * OVER duplique C$
                        * SIZE -> syst.binary
DUP3PICK                * DUP long. c$ = L
                        * 3 PICK duplique N
#/                      * division de 2 sytem
                        * binary
                        * résultats
                        * niveau 2 : reste donc
                        * mod (L , N) soit R
                        * niveau 1 : quotient
                        * entier Q donc nombre
                        * de lignes de N caract.
ROTOVER                 * ROT niveau 1 : L
                        * OVER duplique Q
#+                      * addition de 2 system
                        * binary
                        * ici L + Q ( Q est aussi
                        * le nombre de newline à
                        * ajouter
#2*                     * multiplication de
                        * 2 system binary
                        * ici nombre de caract.
                        * = 2 -> n quartets
NULL$SWAP               * NULL$ -> ""
                        * SWAP  -> n
EXPAND                  * réserve pour une chaine
                        * de n quartets soit 2n
                        * un espce mémoire
4UNROLL                 * 4 ROLLD
                        * niveau 5 : R
                        * niveau 4 : N
                        * niveau 3 : N
                        * niveau 2 : R
                        * niveau 5 : Q

CODE                    * indique au compilateur
                        * un bloc assembly code
                        * jusqu'à ENDCODE
          gosbvl  =POP2#  * prend 2 system binary
                          * niveau 2 -> A(A)
                          * niveau 1 -> C(A)
                          * et DROP2
          r0=c            * sauve Q nombre de
                          * lignes entières
          r1=a            * sauve R longueur
                          * ligne non entière
          gosbvl  =POP#   * prend 1 system binary
                          * niveau 1 -> A(A)
                          * et DROP
                          * N longueur ligne
          a=a-1    a      * pour compteur de carac.
          ar0ex           * sauve N - 1 -> R0
                          * A(A):=Q
          gosbvl  =SAVPTR * sauve les pointeurs
          c=dat1  a       * adresse mémoire
          d0=c            * réservée
          rstk=c          * sauve sur la pile
          d0=d0+  10      * début données
```

```
        d1=d1+  5       * passe [redacted]
        c=dat1  a       * adresse chaine de codes
        d1=c            * pointe C$
        d1=d1+  10      * début des codes
        b=a     a       * compteur lignes
        lchex   0A      * caract. newline
        d=c     a       * [redacted] en D(B)
        gonc    decr    * saute en decr
                        * pour tester s'il y a
                        * [redacted] ligne entière
loop                    * transfert et ajout
                        * de newline
        a=r0            * compteur nombre de
                        * caractères
loop2                   * transfert de lignes
                        * entières
        c=dat1  2       * transfert caractère
        dat0=c  2       * par caractère
        d1=d1+  2       * ( 2 quartets )
        d0=d0+  2       *
        a=a-1   b       * compteur de caractères
        gonc    loop2   * encore des caractères
        c=d     b       * sinon on termine la
        dat0=c  2       * ligne par 1 newline
        d0=d0+  2       *
decr    b=b-1   a       * compteur lignes
        gonc    loop    * encore des lignes
        a=r1            * sinon A(A):=R
        a=a-1   a       * actualise le compteur
        goc     out     * sortie si R = 0
loop3                   * transfert de ligne
                        * non entière
        c=dat1  2
        dat0=c  2
        d0=d0+  2
        d1=d1+  [redacted]
        a=a-1   b       * compteur de caractères
        gonc    loop3   * encore des caractères
out     gosbvl  =GETPTR * sinon restaure les
                        * pointeurs
        d1=d1+  5       * pointe niveau 2
        c=rstk          * récupère adresse de [redacted]
        dat1=c  a       * adresse -> 2ième niveau
        d1=d1-  5       * pointe 1er niveau
        govlng  #05149  * actualise PC retour RPL

ENDCODE                 * marque la fin du bloc
                        * assembly code et retour
                        * mode RPL de compilation
        DROP            * DROP sur la chaine C$
    ;                   * marqueur de fin corres-
                        * :: après FORTYNINE
    ;                   * marqueur de fin pour le
                        * programme entier
```

Conclusion: cet article n'a eu pour but que de donner un aperçu de la programmation en *RPL* et d'inciter le plus possible de programmeurs à en faire usage.

NB : le compilateur n'accepte les commentaires que pour les parties non RPL.

Guy Toublanc (276)

UN SYSTEME D'AIDE POUR HP48

Tout le monde [redacted] besoin d'aide lorsqu'il utilise un ordinateur de temps en temps, que ce soit un ordinateur de bureau, portable ou de poche. La HP 110 fut un exemple précoce d'une machine avec un software et une aide en lignes installés dans la machine. Des ordinateurs qui tiennent dans la main tels que la HP95 LX ont une aide en ligne pour la plupart software intégré. La HP48 n'en a pas.

La valeur de cette aide intégrée dans la ROM est un sujet que l'on peut débattre. Cependant peu de personnes nieront son utilité lorque cela est fournie sous une forme électronique.

Aujourd'hui, le matériel de poche comme la HP48 est de loin plus compliquée que les premières programmables telles que la HP-65 et la HP-41. Avec 256 Ko de software intégré, et une mémoire vive de plusieurs centaines de Ko, les fonctions et les programmes ont grandi en taille et en capacité. La HP28 C fût le premier exemple majeur où la documentation dépassait de loin la taille et le poids de la machine, et de nombreux utilisateurs se plaignirent. A côté de ceux qui se plaignaient de la taille des manuels, il y avait ceux qui se plaignaient du manque de détails et d'exemples spécifiques de programmation. Hewlett-Packard essaya d'établir un compromis entre le volume et la précision et fit raisonnablement du bon travail.

Alors que les ordinateurs de poche et les 'palmtops' deviennent de plus en plus puissants, nous avons besoin de solutions pour de meilleures documentations. Transporter un manuel de la taille d'un annuaire téléphonique n'est pas acceptable. Travailler avec un ordinateur de poche ou un ordinateur personnel signifie que la machine (et la facilité de l'utiliser) est toujours avec l'utilisateur. La facilité à trouver rapidement une référence [redacted] une fonction ou un programme utilisé occasionnellement est importante. Un bon index nous aide beaucoup ainsi que des exemples. Mais je ne peux, ni [redacted] veux transporter un manuel. Je suis paresseux. Je veux juste le faire, en apprenant comme je vais. Des messages d'erreurs aident, mais avoir un beep

d'erreur me gêne. J'ai besoin d'une aide en ligne pour quelques unes des fonctions les plus ésotériques, des bibliothèques avec une aide prévue, ou peut-être utiliser une aide que l'on peut installer...

C'est alors que naquit l'idée de fournir un système d'aide configurable pour l'utilisateur dans la HP48. Des concepts réalisés dans le système sont les suivants :

- *Une aide disponible à partir du clavier, provenant du programme, ou même lorsque l'on touche une clé (ligne de commande active).* Pouvoir demander de l'aide à n'importe quel moment si j'en ai besoin.
- *Passer au-dessus de n'importe quelle aide intégrée.* Si je souhaite fournir ma propre aide pour quelque chose, laissez-moi faire. Certains des modules de la HP 41, tels que le ROM plotter le permettait en utilisant des XEQs globaux au lieu de compiler des références de XROM où la flexibilité était appropriée.
L'aide locale. Une ROM pour faire de la navigation céleste pourrait inclure des informations sur quelques unes des fonctions ou des données. Un répertoire de programmes pour le dimensionnement de canalisations pourrait inclure une aide sur les conversions de viscosité et sur les flux de fluide.

LE SYSTEME D'AIDE DE LA HP 48

Le système inclus ici est simplement nommé ?, pour faciliter la frappe (alpha, orange shiftée flèchearrière). Des répertoires d'aide nommés %? sont stockés dans des emplacements appropriés tels que :

```
HOME ----+---- MATH ----+---- CALC ----+---- %?
         |              |              +---- RUNGE
         |              |
         |              +---- ALGBR -------- %?
         |
         |
         +---- GAMES -------- CHIP
         |
         +---- %?
```

METHODE D'ACCES

Dans la structure de la mémoire ci-dessus les répertoires MATH, GAMES et %? sont dans la racine HOME. Le répertoire MATH contient les sous-répertoires CALC et ALGBR, dont chacun possède un répertoire local d'aide %?. Si vous travaillez dans le répertoire CALC ou dans RUNGE? et spécifies un nom et évalues ?, ? regardera le répertoire %? (par exemple, où le chemin est HOME MATH CALC %? pour une variable par ce nom). Une limitation courante du système est que l'aide globale pour des variables utilisateurs n'est pas disponible.

Un algorithme décrit mieux la séquence cherchée :

```
      Début
        |
   Aide disponible
   dans répertoire     Oui
   local d'aide ? ------------
       (%?)                   |
        |                     |
        |   Non               |
        |                     |
  Aide disponible             |
   dans un autre
 répertoire d'aide,   Oui    Rappeler et
recherchant en amont ----> évaluer l'objet -----
   dans le chemin ?            d'aide            |
        |                                        |
        |   Non                                  |
        |                                        |
  Aide disponible                                |
    dans une                                     |
  bibliothèque ?      Oui    Evaluer l'aide      |
  (Fonction XLIB    ----> (en bibliothèque) --> |
  nommée '?nom')                                 |
        |                                        |
        |   Non                                  |
        |                                        |
  Intégrée dans                                  |
 la fonction ou       Oui                        |
dans la commande ? ------>    Evaluer 'USAG' --> |
        |                                        |
        |   Non                                  |
        ----------------------------------------> retour
```

LIMITATIONS COURANTES

A cause de l'attitude inhabituelle du programme USAG lorsqu'il est évalué sous le contrôle d'un programme, le système d'aide ne doit pas inclure des appels à ce programme. Il peut être tout à fait possible d'ajouter cette caractéristique souhaitée à l'avenir.

Une autre limitation est que seule un répertoire d'aide est recherché (pour une objet d'aide, souvent une chaîne alphanumérique ou un programme) plutôt que d'avoir accès à d'autres objets d'aide dans d'autres répertoires tels que le répertoire HOME. L'aide 'globale' (localisée dans le répertoire HOME %?, serait facile à ajouter au programme CKID (ci-dessous) si souhaité quoique le temps de recherche augmente.

PROGRAMMES

Le coeur du système est le programme Aide appelé '?'. l'entrée requise est : 'nom' ou "nom" ou { nom } (note: 'Nom' n'est utilisé que pour des variables (utilisateur) globales).

Listing (en système RPL)

```
::  CK1NoBlame                    vérifie arguments
    CK&DISPATCH1
    # 3h                          si chaîne entrée
   :: "{" ▮▮ DOSTR> ID 'CKLIST'   créer { nom } et
                                  lancer CKLIST
   ;
    # 5h ID 'CKLIST'              si liste entrée,
                                  aller à CKLIST
    # ▮▮
   :: ONE{}N ID 'CKLIST'          si nom global 1
                                  mettre dans list
   ;                              et aller à CKLIST
    # 0h ID '?ERR'                autrement donner
                                  le message erreur
;
```

Voici la forme ASC de '?' :

```
%%HP: T(1)A(R)F(.);
"47A2084E2010F3D9D20D29512BF81D0040D9D2067156391
507314184E206034B4C4943545B21301204084E206034B4C
4943545B2040D9D20DEE3284E206034B4C4943545B2130FE
F3084E2040F3542525B2130B21302BAC"
BYTES: 85.5     Checksum: #CAB2h
```

CKLIST

C'est en fait la plupart du travail, appelant d'autres routines qui dépendent du type d'objet dans la liste. L'entrée requise est { nom }. Note: Aucun argument initial vérifiant est fait puisque c'est un sous-programme. Ne pas évaluer sans une liste non-vide dans le niveau 1.

Listing :

```
:: x<< DUP CARCOMP xTYPE          { nom } type
  xRPN-> LAM 'nm' LAM 'nt'
  x<<
  xCASE
   :: LAM 'nt' % 6 x==            si nom global
  xTHENCASE
     :: xIFERR                    voir d'abord dans
                                  répertoire aide
       :: LAM 'nm' ID 'CKID'
       ; xERRTHEN DROP
       xIFEND
     ;
  xIFEND
 LAM 'nt' DUP xDUP ▮ 14 x==         If XLIB
 ▮▮▮ ▮ 18 x== xOR SWAP ▮ 19 x== xOR    ou fonction
                                       ou commande
  xTHENCASE
      :: xIFERR
        :: LAM 'nm' ID 'CKID'     regarde d'abord
                                  dans répertoire
                                                aide
      ;
       xERRTHEN
      :: DROP
         xIFERR
         :: LAM 'nm' ID 'CKXLIB' alors dans
                                 bibliothèque
                                 si non trouvé
         ;
         xERRTHEN DROP
         xIFEND
      ; xIFEND
    ; xIFEND
   ; xIFEND x>>ABND
 x>>
 ;
```

Forme ASC de CLIST :

```
▮▮ HP: T(1)A(R)F(.);
"D9D20E1632881309805068BC11C432D6E2020E6D6D6E2020
E647E1632D8732D9D20D6E2020E647233A2279E18A732D9D2
0FD332D9D20D6E2020E6D684E204034B49444B2130F173244
2305DF22B21305DF22D6E2020E64788130788F115CC1279E1
322303392010000000000000810279E1908E13223033920100
000000000910279E1908E18A732D9D20FD332D9D20D6E202
0E6D684E204034B49444B2130F1732D9D2044230FD332D9D2
0D6E2020E6D684E206034B485C49424B2130F1732442305DF
22B21305DF22B21305DF22B21305DF22EF53293632B2130D
111"
BYTES: 219     Checksum:  #111Dh
```

CKID

Ce programme vérifie si l'aide est disponible dans un répertoire d'aide sinon renvoie un message d'erreur et retourne au programme appelant. Entrée requise est { nom }.
Note : aucune vérification d'argument initial n'est faite puisque c'est un sous-programme. Ne pas évaluer sans une liste non-vide dans le niveau 1 !

Listing

```
:: xPATH ID '%?'                Rappelle le
                                  chemin courant
                                  et va dans le
                                  répertoire aide.

  SWAP CARCOMP
 xIF DUP xTYPE ▮ 6 x#?            si 'nom' n'est
                                  pas un nom
                                  global
 xTHEN
 :: "{" SWAP DO>STR ▮▮ DOSTR> CARCOMP  Si l'objet
                                  d'aide ayant le
                                  même nom n'est
```

AH ! VOUS ECRIVEZ

Vous vous sentez en verve, mais vous ne savez pas sous quelle forme "l'équipe de rédaction" souhaite recevoir votre prose. C'est ici que se trouvent les réponses à vos questions.

Dans la mesure du possible, vous devez nous envoyer vos écrits sur support magnétique (carte, cassette ou disquette) dans le format lisible directement pour votre machine. Vous pouvez taper vos articles sur IBM PC, mais n'utilisez pas de traitement de texte (WORD...). Utilisez plutôt un éditeur simple manipulant des fichiers en ASCII pur, sans codes d'enrichissement. Soyez sans crainte, nous vous retournerons vos biens après copie.

Si vous n'avez pas accès à un IBM, et seulement dans ce cas là, vous pouvez à la rigueur nous envoyer des disquettes provenant de MacIntosh.

Si vous ne pouvez pas utiliser de support magnétique, ou ne pouvez vous rendre aux réunions, alors et alors seulement faites le sur papier.

Que ce soit sur une feuille de papier, ou sur support magnétique, ne dépassez pas 50 caractères par ligne.

Pour nous épargner du travail, insérez dans votre texte les commandes de formattage suivantes (et non les commandes du formatteur HP-71) :

"^" centre un titre, par exemple :
^TITRE

"\" (CHR$(92)) marque le début et la fin d'un paragraphe. Par exemple :

\Début de paragraphe exprimant le contenu de vos idées qui, même si vous en doutez, intéressera certains des membres du Club. Surtout si vous vous sentez débutant. Les articles pour débutants écrits par des débutants sont ceux qui manquent le plus. Fin de paragraphe.\

Pour écrire une accolade ({ ou}), il faut doubler l'accolade, une accolade simple ayant une signification spéciale dans l'édition de JPC.

Les utilisateurs de HP-71 utiliseront le fichier CHARLEX, qui a été souvent listé dans le coin des Lhex, pour utiliser les caractères accentués du jeu Roman8.

Les utilisateurs de HP-48 doivent nous envoyer les programmes RPL sous deux formes : Binary, plus ASCII avec *Translate Code* : 3. Les programmes composés de plusieurs variables doivent être dans un répertoire à transférer sur disquette et dans les deux modes. Si vous insérez dans le corps des articles le texte des programmes, veuillez faire précéder ce texte de `{rpl` et le faire suivre de `{endrpl`.

Les utilisateurs de HP-95 et MS-DOS nous transmettront les sources de leurs programmes accompagés des fichiers compilés. Il nous indiquerons le nom et la version du compilateur ou de l'interpréteur qu'ils ont utilisé.

Le Bureau

PPC PARIS SE REUNIT
UNE FOIS PAR MOIS

Comme vous le savez peut être déjà, PPC Paris se réunit une fois par mois, en plein coeur de Paris. Amenez votre matériel, votre bonne volonté et vos idées ! Plus vous en apporterez, et plus vous en trouverez chez vos collègues de PPC.

Ces réunions se déroulent de manière très libre, aucun ordre du jour, discussion ou autre n'étant imposé. Un membre du bureau est toujours présent. Ainsi, si vous désirez remettre votre article tout frais au Journal, si vous avez des suggestions à faire, si vous voulez vous procurer des anciens numéros de JPC, ce sera en principe toujours possible.

Si donc cela vous intéresse, n'hésitez plus un seul instant, venez nous rejoindre tous les premiers samedis de chaque mois (sauf en période de vacances scolaires) au :

Centre de Jeunesse et de Loisirs Jean Verdier
11 rue de Lancry
75010 Paris

et en montant au deuxième étage, vous entendrez des éclats de rire et des discussions passionnées vers la salle 215. Attention, toutefois, de venir entre 16 et 19h.

Pour l'accès en métro, trois possibilités s'offrent à vous :
- Métro Strasbourg Saint Denis :
Sortie porte St Martin / Bd St Denis, coté pairs
- Métro République :
Sortie Bd St Martin, coté pairs
- Métro Jacques Bonsergent :
Sortie Bd Magenta, coté impairs.

Ah, j'oubliais ! JPC est (souvent) distribué en avant première lors de ces réunions... A bon entendeur, salut !

Les dates des prochaines réunions sont :

Samedi 3 Octobre 1992
Samedi 7 Novembre 1992
Samedi 5 Décembre 1992
Samedi 9 Janvier 1993
Samedi 6 Février 1993
Samedi 6 Mars 1993
Samedi 3 Avril 1993
Samedi 5 Juin 1993

NOUS EN AVONS

La coopérative du Club vous propose :

- Des **anciens numéros** de JPC (Préciser les numéros).
- Des **années complètes** de numéros de JPC (février à janvier).
- Des **manuels de service** du HP-41 et du HP-75.
- La **Programathèque HP-71**, regroupant tous les Lex et programmes pour HP-71 et HP-75 parus à ce jour dans *JPC*. Elle vous est livrée avec un catalogue décrivant brièvement tous les programmes.
- La **JPC Rom** pour HP-71, comportant plus d'une centaine de fonctions en assembleur (mathématiques, programmation structurée, utilitaires systèmes...). Joindre une Eprom vierge (64 Ko) à votre règlement. Il est possible d'y ajouter vos propres programmes.
- Les **utilitaires de développement** pour HP-48 (fonctionnant sur IBM PC), développés mais non supportés par HP. La disquette contient un Assembleur et un Linker pour le microprocesseur Saturn, un compilateur RPL et un générateur de librairies. Les fichiers également inclus dans la disquette comprennent les notices de ces logiciels, le jeu d'instruction du microprocesseur, la description de la structure logicielle de la HP-48, du langage RPL, ainsi que le listing de plus de 2000 points d'entrées. Vous pouvez obtenir cette documentation imprimée et reliée par nos soins.

Le port, et eventuellement la disquette (précisez le format) sont inclus dans les prix.

Si vous souhaitez des renseignements complémentaires, n'hésitez pas à nous contacter.

VOUS EN VOULEZ

Nom :
Prénom :
No d'adhérent :
Adresse :

	Quantité	Prix Unitaire Adhérent	Prix Unitaire Autre	Prix Total
Anciens numéros de JPC	x	40 FF	45 FF	
Année complète de JPC	x	300 FF	350 FF	
Manuel de service pour HP-41	x	60 FF	75 FF	
Manuel de service pour HP-75	x	60 FF	75 FF	
Programmathèque HP-71	x	60 FF	75 FF	
JPC Rom + Manuel	x	500 FF	600 FF	
JPC Rom + Manuel + vos propres programmes	x	700 FF	800 FF	
Actualisation Eprom	x	150 FF	200 FF	
Kit de développement pour HP-48 (disquette seule)	x	40 FF	50 FF	
Kit de développement pour HP-48 (disquette + doc imprimée)	x	170 FF	200 FF	
			Total	FF

Préciser éventuellement les numéros de JPC commandés :

PPC PARIS

Association régie par la loi de 1901, enregistrée
à Paris le 2 décembre 1982 sous le numéro 82/3240

BULLETIN D'ADHESION

Nom |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

Prénom |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| Date de naissance |_|_|/|_|_|/|_|_|

Adresse |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

Code Postal |_|_|_|_|_| Ville |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

Téléphone Domicile |_|_|_|_|_|_|_|_|_|_|_|_|_|_| Bureau |_|_|_|_|_|_|_|_|_|_|_|_|_|

Profession ______________________________

Intérêts ______________________________

____

Matériel HP en votre possession ______________________________

____

Autre matériel informatique ______________________________

____

Comment avez-vous connu PPC Paris ? ______________________________

____

Que recherchez-vous au sein de PPC Paris ? ______________________________

____

____

La Loi No 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, garantit à toute personne justifiant de son identité un droit d'accès et de rectification auprès des services ou organismes chargés de mettre en oeuvre des traitements informatiques comportant des informations nominatives le concernant.

Je souhaite adhérer au club PPC Paris conformément aux statuts de l'Association. Au mieux de ma connaissance, je déclare avoir le droit de fournir tous les programmes et informations que je vous enverrai (sans enfreindre des obligations de secret à l'égard d'autres personnes ou organismes) pour publication dans le Journal de liaison, sans obligations ni responsabilité d'aucune sorte (en cas d'utilisation frauduleuse) de la part des dirigeants de PPC-Paris.

Date |_|_|/|_|_|/19|_|_|

Signature, précédée de la mention "Lu et Approuvé" ______________________________

Le montant de la cotisation s'élève à 350.00 F pour un an.
Etudiants: 300.00 F (justificatif indispensable)

Paiement à l'ordre de "PPC Paris" par chèque bancaire ou virement postal (CCP No 18 823 40 C à Paris). Ne pas utiliser d'Eurochèques.

Veuillez envoyer toute correspondance à :
PPC Paris, BP 604, 75028 Paris Cedex 01, France

```
                                  pas dans le
                                  répertoire aide,
                                  alors erreur
                                  et exit.
      xIF DUP xTYPE ▒ 6 x#?
      xTHEN
      :: SWAP xEVAL # 6d DO#EXIT
      ;
      xELSE
      :: { ID '%?' } ▒WAP x+ ▒WAP   Ou bien
                                  rappeler,
                                  retourner et
                                  évaluer.
        xEVAL xRCL xEVAL
      ;
      xIFEND
  ;
  xELSE                           Ou essayer de
                                  le rappeler,
                                  retourner et
                                  l'évaluer.

  :: { ID '%?' } SWAP x+ SWAP
    xEVAL xRCL xEVAL
  ;
  xIFEND
;
```

Forme ASC de CKID :

```
%%HP: T(1)A(R)F(.);
"D9D20521A184E202052F332230980503CE228813068BC12
33A2D9AE1AFE22D9D20671563223088041391507314198 05
03CE228813068BC1233A2D9AE1AFE22D9D2032230EB3A1B2
040F2051B21305BF22D9D2047A2084E202052F3B21303223
076BA132230EB3A104B02EB3A1B21305DF22B21305BF22D9
D2047A2084E202052F3B21303223076BA132230EB3A104B0
2EB3A1B21305DF22B21303D5D"
BYTES: 154   Checksum: #D5D3h
```

CKXLIB

Ce programme essaye d'éxecuter une fonction bibliothèque nommée de la même façon que 'nom' sauf avec un préfixe '?'. C'est nécessaire parce que les bibliothèque ont un modèle de mémoire plate (par exemple, aucun sous-répertoire tels que des répertoires d'aide), d'où cette convention de nom.

Même précaution que ci-dessus. Aucune vérification d'argument n'est faite!

Listing :

```
:: CARCOMP "{ ?" SWAP DO>STR      Intégré
                                  "{ ?nom }" et
                                  convertir vers
                                  l'objet ▒ liste
   ▒▒ DOSTR> CARCOMP
   DUP xTYPE ▒ 14 x#?             Si l'objet
                                  n'est ▒▒▒ ▒▒
                                  XLIB
   xIF
   xTHEN
   :: # 6h DO#EXIT                Erreur (et
                                  retour).
   ;
   xELSE xEVAL                    sinon l'évalue
   xIFEND
;
```

Forme ASC de CKXLIB :

```
%%HP: T(1)A(R)F(.);
"D9D2098050C2A20B0000B702F332230880413915073141 9
80508813068BC115CC1D9AE13CE22AFE22D9D20B2040F205
1B21305BF22EB3A15DF22B2130F18F"
@ BYTES: 60.5   Checksum: #F81Fh
```

?ERR

Routine erreur de l'aide.

```
::
C$ 42 Input req'd: 'NOM' ,
  { ▒▒▒ }, ou "NOM"
DO$EXIT
;
```

Forme ASC de '?ERR' :

```
%%HP: T(1)A(R)F(.);
"D9D20C2A209500094E607574702275617724 6A30272
E414D45472C2A00202B702E414D45402D7C202F62702
22E414D4542284051B21300D35"
BYTES: 54.5   Checksum: #53D0h
```

L'ASSIGNATION DE LA TOUCHE AIDE

Avec cette assignation de touche, ▒▒▒▒ pouvez accéder à l'aide sur la barre de menu proposé ou taper le nom. C'est une reprise type d'assignation donnant instantannément accès à l'aide, même avec une ligne de commande. (Note je pense q'une modification du code est nécessaire pour invalider ceci). J'ai assigné ceci à la touche bleue SHIFT DEL (normalement utilisée pour la commande dangereuse CLVAR).

Listing

```
:: TakeOver # 1h
  xDO DROP
   WaitForKey UnLockAlpha          Attendre la pression
                                   d'une touche de la
                                   ligne supérieure ou
                                   de la touche ou ENTER
```

```
 DROP UNCOERCE DUPDUP
 % 7 x<? xSWAP % 25 x== xOR
xUNTIL
xENDDO
xIF DUP % 7 x<?                 Si une touche de
                                la rangée haute
                                a été pressée
xTHEN
:: COERCE GETDF                 Obtenir la chaine
                                de l'étiquette de
                                la touche de menu
   CK1&Dispatch                 Si l'étiquette de
                                la touche est un
                                menu (comme
                                SOLVR),obtenir
                                la chaine (2°
                                objet programme)
                                sinon OK, ne
                                faites plus rien
  #8h
  :: CDRCOMP CARCOMP
  ; # 0h NOP
;
xELSE                           D'autre part la
                                touche ENTER a été
                                pressée,alors
                                arrêter pour un
                                nom
:: DROP "Nom?"
   { "{  }" % 2 ID '\Ga' ID 'V' } et convertit
                                  "{ nom }"
                                  en une liste
                                  contenant
                                  n'importe quel
                                  type d'objet nom
   xINPUT DOSTR>
;
xIFEND
ID '?'                          OK, c'est fait!
;
```

Forme ASC pour l'assignation d'une touche d'aide.

```
%%HP:  T(1)A(R)F(.);
"D9D20887049FF303C0324423056F1493D0444230FBD819B
C26743A2EBBE1DBBF102DC1279E1908E1DE0329B6323CE22
88130743A2EBBE1AFE22D9D20AEC8114A40ECE81F3040D9D
203515098050B2130FEF30E8E60B21305BF22D9D2044230C
2A20F0000E416D656F347A20C2A20D0000B70202D7ED2A28
4E2010C884E201065B2130AC42273141B21305DF2284E201
0F3B2130242F"
@ BYTES: 147.5   Checksum: #F242h
```

EXEMPLES

Le temps passe très vite, aussi je donnerai quelques idées pour des objets aide :

1. Des chaînes alpha. retourner une ou plusieurs chaînes de texte dans la pile. Cela servira pour l'aide sans une ligne de commande.
2. Graphics. GROBs occupe de l'espace mais peut être utile. Si en faisant cela vous souhaitez utiliser PGROB et UPGROB (assembler et décomposer GROB) ou les outils RF (éliminer la redondance) pour compresser ou décompresser. La carte HP Equation Library est un excellent exemple d'utilisation de graphiques, avec l'option PIC.
3. Programmes. Ceux-ci peuvent être ce que vous voulez, mais parmi ceux qui viennent à l'esprit utilisons l'excellent programme DLIST trouvé dans le disquette n° 3 des Goodies :

```
%%HP:T(1);
@ DLIST de Joseph Horn
@ Affiche rapidement une liste de neuf chaines en
@ ordre inversé. Elles sont toutes affichées
@ dans la plus petite fonte LCD (taille 1)
@ DANGER ! Utiliser seulement quand le niveau 1
@ contient une liste de chaînes;
@ C. à d. { "Last Line" "Middle Line" "First Line" }.
@ Lancer DLIST avec d'autres types de paramètres
@ peut corrompre la mémoire
@ Ne l'utiliser que dans un programme, quand le
@ contenu de la pile est connue.

"D9D20CA031B7FC19C2A2DBBF10A132D6E2010E6FEF30D6E
2010E69C2A290DA1233A2EEDA17DC81E0CF108F110B821C4
232F2293B21301D5E"
```

Qu'y aura t-il après ?

Plusieurs modifications de ce qui est au dessus sont possibles, mais ces programmes illustrent une méthode permettant de fournir une aide à l'utilisateur qui soit toujours disponible à la pression d'une touche et qui soit aussi compréhensible et modifiable aux souhaits de l'utilisateur. Les calculateurs Hewlett-Packard ont toujours eu une bonne documentation, encore que l'imposant volume des manuels empêche un utilisateur mobile de les transporter constament pour référence. Savoir simplement comment marche une fonction demande plus qu'un aide mémoire. Par exemple, quels sont les paramètres optionnels associés à la commande INPUT ? J'aurais besoin de transporter le volume II du manuel (heureusement que cela n'est pas dans le numéro I) pour trouver cette information, et il n'est pas facile d'y retrouver l'information. Plusieurs bons livres sont disponibles sur la HP48, chacuns couvrant une facette des applications ou du fonctionnement. Les fonctions les plus usuelles devraient être expliquées à différents niveaux directement dans la machine. Utiliser certains logiciels sans avoir à lire le manuel est devenu chose courante. Je veux avoir un certain travail fait directement;ceci étant réalisé, j'en

obtiens plus qu'en travaillant avec le manuel et machine pour découvrir toutes les choses merveilleuses qu'elle peut faire; La période d'apprentissage de la HP48 est en fait très longue mais aussi très fructueuse. La plupart des utilisateurs ne sont pas assez patients, ils peuvent néanmoins maitriser la HP-41 facilement. Mais une comparaison avec la HP-41 n'est pas juste car elle n'a que 12 Ko ou 24 Ko de ROM, une taille écran réduite et pas de possibilités graphiques.

Quelle que soit la forme future de la documentation, nous pouvons être certains qu'elle sera insuffisante. Il est classique de dire que la ROM est complète. Peut-être qu'avec le progrès des mémoires flash et des petits disques durs tels que KittyHawk disponibles pour des ordinateurs de poche nous aurons une profusion d'aide par touches sensitives... Pour le moment nous avons besoin de manuels concis, complets et pourvus d'exemples, avec une bonne indexation ainsi qu'un certain type d'aide en ligne personnalisée par l'utilisateur et facilement installable. J'espère que ceci propose des idées. Que penser d'un répertoire pour HP48 personnalisé et contenant un réperoire d'aide local indiquant quels sont les programmes et quel est leur mode d'emploi?. Tout ce que vous auriez à faire est de d'entrer le nom du programme et d'appuyer sur la touche d'aide. Avoir ce genre d'outil enrichirait agréablement la documentation papier sans la remplacer.

Brian Walsh
Traduit par B. & B. Pougon

UN AUTRE INDEX POUR LE MANUEL HP-48

Revenons le 29 mars 1992, vers la fin du deuxième jour de la conférence des utilisateurs HP à Philadelphie, sur une discussion qui fut soulevée au sujet du Manuel Utilisateur du HP48 et de sa table d'index. En dépit de sa grande taille (29 pages et approximativement deux mille entrées), il fut souligné qu'il restait des mots clefs importants qui avaient été oubliés. Le premier exemple donné fut l'absence d'une entrée pour le séparateur décimal à l'écran, alors qu'il en faudrait une pour savoir que la commande "FM", permet de choisir le type de symbole. (Pour être juste envers l'index HP, il y a effectivement les entrées "Le point comme marque décimale" et "La virgule comme marque décimale").

Ayant fait quelques tables d'index annuelles pour le journal PPC dans le passé, je me suis désigné pour compiler et réviser les index des deux volumes du manuel HP48. Le but était de développer les références indépendamment de l'index original. Comme résultat, l'index HP fut vérifié par recoupement une fois le document virtuellement terminé. Dans la première lecture du manuel de la HP48, les sujets clefs furent compilés dans une liste manuscrite dans l'ordre où elles furent rencontrées. La deuxième passe consista à entrer ces sujets sous un éditeur de texte dans l'ordre alphabétique. Si un sujet avait plusieurs mots clefs, les entrées furent incluses plusieurs fois comme nécessaire. Enfin, des sous-listes d'index majeures furent élaborées comme "Commandes", "Exemples", "Illustrations" et "Menus".

Dans la discussion de mars dernier, il fut suggéré qu'une "bonne table d'index" devrait contenir une moyenne d'au moins deux références pour chaque pages du document source. La table d'index originelle du manuel de la HP48 surpasse cette estimation puisqu'il y a deux mille entrées pour 821 pages de texte. La nouvelle table d'index contient environ entre 3500 et 4000 entrées. On espère que les utilisateurs de HP48 la trouveront comme une ressource utile pour accéder rapidement à un sujet, peut-être même mieux que dans celle de Hewlett-Packard.

Récemment, un manuel révisé d'un seul volume est apparu dans les boîtes de HP48. A la conférence de Philadelphie nous avions été informés par Dennis York que cette version avait été mise à jour de proche en proche, avec l'ajout significatif de l'arborescence des menus dans un des appendices. Peut-être que la table d'index suppléante devrait contenir les références à la fois pour les versions un manuel et deux manuels de la documentation de la HP48. Ceci en considérant la possible incorporation d'autres textes de référence utiles pour HP48 comme futur projet. Tous commentaires ou suggestions sur ce projet sont ardemment bienvenus.

Jake Schwartz
Traduit par Jean-Louis Attenoux (83)

NDLR : Ecrit par un américain, cet index correspond évidement à la version anglaise du manuel. Cependant, les personnes intéressées peuvent évidement en la demander, en attendant qu'une bonne âme ait le courage de la traduire...

INITIATION AU LANGAGE MACHINE

Cet article s'adresse aux débutants qui n'ont jamais osé s'aventurer dans le monde merveilleux du langage machine de la HP 48.

Bien des auteurs ont déja décrit des programmes en LM, mais les explications fournies sont souvent succintes et de fait ils s'adressent à des programmeurs LM confirmés.

C'est pourquoi j'ai décidé d'écrire quelques lignes de LM avec force explications.

Mais pour bien assimiler ce qui va suivre il est nécessaire de posséder un support écrit tel le livre "Voyage au centre de la HP48" de P. Courbis et S. Lalande (éd. ANGKOR). Tout ce qui a été écrit ici repose sur la lecture de ce livre.

Commençons par le plus simple: la lecture d'une donnée située au niveau 1 de la pile. Ecrivons un "E" dans la pile.

On sait que cette chaîne sera représentée en mémoire par `C2A207000054`. De plus la machine ne stocke pas directement cet objet dans la pile; elle stocke seulement l'adresse de celui-ci. Ceci est avantageux car il est facile de gérer une pile où les objets font toujours la même longueur : ici 5 quartets.

Pour accéder à cet objet il faut agir en indirect comme existe l'adressage indirect sur la HP41. C'est le registre D1 qui contient l'adresse du niveau 1 de la pile. Il faut donc charger la donnée de cette adresse dans un registre interne: A par exemple. Cela donne l'instruction LM :

```
A=DAT1 A
```

Le premier A est un des registres internes de la HP48. On l'oblige à égaler non pas D1, mais la donnée (qui est une adresse) que D1 contient (`=DAT1`; style indirect) . Cette donnée est l'adresse de l'objet situé au niveau 1 de la pile. Le 2° A signifie qu'on charge cette donnée (l'adresse de l'objet situé au niveau 1 de la pile) dans le champ A du registre A. En effet cela suffit puisqu'une adresse fait 5 quartets et que le champ A a une longueur de 5 quartets.

Or une chaine en représentation interne contient un prologue (5 quartets: C2A20) suivi de sa longueur en quartets + 5 (5 quartets: 70000). Il faut donc sauter ces informations pour accéder à la donnée (E; 2 quartets: 54) proprement dite.

Cela s'effectuera par 2 instructions LM:

```
D1=A
D1=D1+ 10
```

Le registre A a été précédemment chargé par l'adresse de la donnée. On charge D1 avec cette adresse (`D1=A`; style direct). Puis on incrémente D1 de 10, ceci nous faisant sauter les quartets nécessaires (5+5) pour accéder à l'adresse de la donnée: E (on travaille toujours sur des adresses pour le moment). Il suffit maintenant d'aller lire le contenu de l'adresse ainsi obtenue. Ceci est fait avec l'instruction en LM suivante :

```
A=DAT1 B
```

On travaille toujours avec le registre A. On le charge avec la donnée située à l'adresse pointée par D1 (`=DAT1`; style indirect). Cette donnée est stockée dans le champ B de A: cela suffit puisque la donnée fait 2 quartets de long et que B fait précisément cette longueur. Le champ B de A contient en définitive 54, soit la lettre E. Tout est prêt maintenant pour que l'on puisse travailler.

EXEMPLE :

Changeons la chaîne "E" située au niveau 1 de la pile en "F".
Rien de plus facile puisque nous savons lire une telle donnée. Il suffit d'ajouter à ce que nous venons de voir les instructions LM suivantes:

```
A=A+1  B
DAT1=A B
```

Nous avons chargé précédemment le champ B du registre A avec 54 (E). Avec la 1° instruction on fait passer la valeur du champ B du registre A à 64 (on se rappelle que le microprocesseur SATURN équipant la HP48 retourne les quartets en mémoire; ainsi le nombre 54 en mémoire correspond en réalité à 45; quand on lui ajoute 1 cela donne 46 puis quand on le remet en mémoire on retrouve 64). Ceci correspond au code ASCII (inversé) de la lettre F. Avec la 2° instruction on fait le contraire de ce que nous avons vu tout à l'heure, à savoir charger indirectement D1. D1 contient une adresse; on met à cette adresse la nouvelle donnée (contenue dans le champ B du registre A) et le tour est joué (`DAT1=A B`; style indirect).
L'instruction `D1=A B` n'aurait pas marché, puisqu'on aurait modifié l'adresse qui référençait la donnée (F).

Nous pouvons donc écrire notre premier programme en LM :

```
CCD20            prol. code
*****            5 quart. de long. code
GOSBVL 0679B     adr. sauv. reg.
A=DAT1 A
D1=A
D1=D1+ 10
A=DAT1 B
A=A+1  B
DAT1=A B
GOSBVL 067D2     adr. charg. reg.
A=DAT0 A         les 3 inst. suiv.
D0=D0+ 5         cherchent A
PC=(A)           retourner au RPL
```

ATTENTION :

Comme tout programme écrit en LM, il faut faire excessivement attention à tout ce qu'on fait: la moindre erreur pardonne rarement. Ici par exemple, lancer le programme sans mettre de donnée au niveau 1 de la pile, ou mettre autre chose qu'une chaîne provoquerait un beau plantage.

Voici pour ce programme la chaîne de codes à assembler soit avec le programme de P.C. et S.L. dans leur livre cité plus haut, soit avec ASSCOD ou ASC→' de G.Toublanc qui ont été publiés dans *JPC n°78*.

```
CCD20 F2000 8FB97 60143 13117 914BB 64149 8F2D7
60142 16480 8C
```

Remarque: pour obtenir la longueur du code (représentée par les 5 * dans le listing il faut compter tous les quartets du listing sauf le prologue, transformer ensuite ce nombre en hexa, l'inverser (car le SATURN inverse les codes) et le compléter à 0 pour obtenir 5 quartets. Pour vous exercer cherchez vous aussi à obtenir la longueur F2000 du listing précédent.

Assemblez la chaine de codes et mettez la en mémoire sous le nom de votre choix. Mettez "E" dans la pile et lancez le programme. Vous devez obtenir "F".

Continuons notre exploration en modifiant légèrement le programme précédent. Nous n'allons plus lire une chaîne composée d'une lettre, mais un réel, comme par exemple le nombre 11. Nous allons en fait essayer d'incrémenter ce nombre de 1.

Pour commencer il faut lire la mantisse de ce nombre. Utilisons la même méthode que précedemment soit le style indirect. En LM on a :

```
A=DAT1 A
D1=A
D1=D1+ 8
A=DAT1 M
```

D1 contient toujours l'adresse du niveau 1 de la pile. Charger le champ A du registre A de manière indirecte (A=DAT1 A) revient à y mettre l'adresse de l'objet situé au niveau de la pile. Ensuite on charge D1 avec cette valeur et on l'incrémente de 8. Cela nous permet d'accéder à la mantisse du réel. En effet un réel est constitué d'un prologue (33920 : 5 quartets) suivi de son exposant (3 quartets). Ainsi incrémenter D1 de 8 (D1=D1+8) nous permet d'arriver à l'adresse de la mantisse (ici 11). Il ne reste plus qu'à charger le registre A indirectement afin d'y mettre la mantisse (l'instruction A=D1 M n'aurait rien donné, seule une adresse y aurait été chargée). Le champ M du registre A est fait spécialement pour y mettre une mantisse, c'est pour cela qu u'on l'utilise ici.

Nous allons maintenant incrémenter la mantisse de 1. Nous allons utiliser le registre C que nous chargerons avec la valeur 010000000000. Quand nous chargeons un registre tel A ou C avec une constante il faut toujours fixer le registre P car celui-ci conditionne le chargement de A ou C. Bien que ceci ne soit pas obligatoire il faut essayer de s'y habituer car cela évite souvent des mauvaises surprises. De plus ici cela s'impose car on travaille sur le champ M qui est situé au milieu des registres A ou C. Cela donne les instructions LM suivantes:

```
P=    3
LCHEX 010000000000
A=A+C M
```

On fixe P à 3 pour charger la constante 010000000000 exactement dans le champ M du registre C, puis on ajoute le champ M de C au champ M de A c'est-à-dire que l'on incrémente la donnée précédemment mise dans le champ M de A (11 dans l'exemple) de 1.

Il ne reste plus qu'à remplacer l'ancienne valeur par la nouvelle. Ceci par l'instruction LM:

```
DAT1=A M
```

Par cette instruction on met en mémoire le champ M de A (style indirect). C'est exactement l'inverse de l'instruction LM précédente:

```
A=DAT1 M
```

Ceci nous donne le programme LM suivant:

```
CCD20
*****
GOSBVL 0679B
```

```
A=DAT1 A
D1=A
D1=D1+ 5
A=DAT1 M
P=     3
LCHEX  010000000000
A=A+C  M
DAT1=A M
GOSBVL 067D2
A=DATO A
DO=DO+ 5
PC=(A)
```

Ce qui nous donne la chaîne de codes suivante:

```
CCD20 14000 8FB97 60143 13117 71535 233B0 00000
00001 0A5A1 5158F 2D760 14216 4808C
```

Assemblez la chaîne, stockez la en mémoire et testez le programme en mettant 11 au 1° niveau de la pile. Lancez le, vous devez obtenir 12. C'est vraiment fantastique, [illegible] a réinventé l'addition de la HP48!

Il y a peut-être une question que vous vous posez en ce moment: pourquoi a-t-on chargé le registre C avec la constante 010000000000? Pour répondre à cette question mettez 100 au niveau 1 de la pile et lancez le programme. Que constatez vous? Pourquoi? Changez maintenant dans la chaîne de codes la valeur de la constante par 001000000000, réassemblez la et remettez la en mémoire. Mettez 11 dans la pile et lancez le programme. Quelle valeur obtenez vous? Pourquoi? Essayez de trouver un programme permettant d'incrémenter de 10,1 le nombre 10000.

Changeons de registre maintenant et intéressons nous à l'écran. L'écran est l'interface de communication entre l'homme et la machine par excellence. Aussi nous allons écrire quelques programmes en LM le concernant.

Dans le premier programme nous allons simplement inverser un quartet de l'écran. Pour cela il nous faut l'adresse de début de l'écran. Celle-ci est conservée en mémoire en #7050Eh (h: hexadécimal). Chargeons le registre D0 avec cette adresse et lisons y la donnée qu'il contient, c'est-à-dire que nous allons lire la donnée du début de l'écran. En LM cela donne:

```
DO=     7050E
A=DATO 5
DO=A
P=     0
A=DATO P
```

On charge D0 avec l'adresse mémoire qui contient l'adresse de début de l'écran (DO= 7050E). Etant donné que D0 contient une adresse, pour accéder à la donnée il va nous falloir lire 2 fois en indirect et nous chargerons le registre A avec cette valeur.

A=DATO 5 : charge l'adresse de l'écran en A champ A (champ A car cela correspond à la taille d'une adresse)
DO=A : met l'adresse de l'écran en D0
P=0 : initialisation du registre P car on s'en sert juste après
A=DATO P : charge la donnée située au début de l'écran à partir du quartet 0 de A (puisque P=0), ceci ne va charger qu'un quartet dans A car D0 contient une adresse et une adresse contient un quartet.

Il suffit ensuite d'inverser la valeur du registre A puis de remplacer l'ancienne valeur par la nouvelle valeur obtenue, ce qui se fait à l'aide des 2 instructions LM suivantes:

```
A=-A-1 P
DATO=A P
```

Remarque : dans ce programme on a utilisé D0 mais on aurait aussi bien pu utiliser D1.
Finalement on obtient le programme LM suivant:

```
CCD20
*****
GOSBVL 0679B
DO=    7050E
A=DATO 5
DO=A
P=     0
A=DATO P
A=-A-1 P
DATO=A P
GOSBVL 067D2
A=DATO A
DO=DO+ 5
PC=(A)
```

La chaîne de codes est :

```
CCD20 73000 8FB97 601BE 05071 42130 20152 088C1
5008F 2D760 14216 4808C
```

Assemblez la chaîne, stockez la et lancez le programme. En haut de l'écran à gauche vous devez voir s'allumer 4 petits points (un quartet) puis s'éteindre aussi rapidement. Ceci s'explique par le fait qu'il faudrait geler l'écran pour les laisser allumés et on ne le fait pas dans ce programme (vous pouvez simplement le faire par le programme RPL: Code 1 Freeze).

Allons un peu plus loin en allumant non plus 1 mais 16 quartets. Nous allons reprendre la même structure que le programme précédent mais nous changerons

notre champ de travail. Nous n'allons plus travailler avec P mais avec le champ W (qui veut dire wide: entier). Les instructions qui changent sont donc celles qui sont en rapport avec P; elles deviennent simplement :

```
A=DAT0 W
A=-A-1 W
DAT0=A W
```

Avec la 1° instruction on charge dans tout le registre A (soit 16 quartets) les données dont la première est pointée par l'adresse située dans D0. (on agit toujours en direct pour avoir accès aux données). Avec la 2° instruction on inverse tout A. Et avec la 3° on remplace la valeur.

Cela donne le programme LM suivant:

```
CCD20
*****
GOSBVL 0679B
D0=    7050E
A=DAT0 A
D0=A
A=DAT0 W
A=-A-1 W
DAT0=A W
GOSBVL 067D2
A=DAT0 A
D0=D0+ 5
PC=(A)
```

Le code donne :

```
CCD20 53000 8FB97 601BE 05071 42130 1527B FC150
78F2D 76014 21648 08C
```

Remarque : on n'a pas eu besoin d'initialiser le champ P ici car on ne s'en sert pas. On aurait pu faire autrement en impliquant P. En effet en le fixant à 15 et en travaillant ensuite avec le champ WP (wideP) on aurait obtenu le même résultat. Essayez. De même avec le programme précédent pour allumer 1 quartet: laissez P à 0 et au lieu de travailler sur P, travaillez sur WP. Vous obtiendrez le même résultat.

Conclusion : en LM il n'y a pas 1 solution mais plusieurs. Généralement ce n'est pas très important car le LM est très rapide. Mais si vous êtes un puriste vous chercherez toujours la solution la plus courte en octets ou la plus rapide en temps. Nous verrons un exemple de ceci tout à l'heure.

Pour l'instant continuons notre initiation en allumant 1 pixel. C'est légèrement plus compliqué. Quoique... En fait nous allons tricher un peu. En effet, comme nous travaillons en mode texte nous savons que le pixel situé en haut à gauche de l'écran est éteint et c'est celui-ci que nous allons allumer.

Il nous faut toujours lire la valeur du quartet en haut à gauche pour pouvoir travailler sur la bonne donnée. On ne peut pas travailler sur n'importe quoi. On va donc prendre la valeur du quartet concerné, changer le bit qui va bien pour allumer notre pixel et recharger le quartet ainsi modifié à l'adresse mémoire de l'écran. Cela donne les instructions LM suivantes:

```
P=     0
A=DAT0 P
ABIT=1 0
DAT0=A P
```

Vous connaissez maintenant 3 instructions sur les 4. La 4° est ABIT=1 0 : elle signifie que vous mettez le bit 0 du registre A à 1. Comme D0 contient l'adresse de l'écran c'est comme si nous allumions notre pixel. Je vous rappelle que faire ça marche ici car on sait très bien que tout le quartet est éteint. Le programme LM donne:

```
CCD20
*****
GOSBVL 0679B
D0=    7050E
A=DAT0 A
D0=A
P=     0
A=DAT0 P
ABIT=1 0
DAT0=A P
GOSBVL 067D2
A=DAT0 A
D0=D0+ 5
PC=(A)
```

Le code est :

```
CCD20 93000 8FB97 601BE 05071 42130 20152 08085
01500 8F2D7 60142 16480 8C
```

Maintenant essayez d' allumer un autre pixel du quartet. C'est très facile:

Il faut agir sur l'instruction ABIT. Pour vous éviter du travail (surtout si vous n'avez pas d'assembleur) prenez l'habitude de stocker votre chaîne de codes hexa avant de l'assembler; cela vous permettra de changer facilement un code sans tout refaire.

Christophe Vaillant (523)

NDLR: nous connaissons les difficultés d'un exposé d'initiation où il faut partir de rien pour expliquer tout. Aussi nous sommes reconnaissant à l'auteur de s'être jeté à l'eau et nous espérons qu'il y aura une suite à cet article.
Nous avons respecté le style imagé de l'auteur et n'avons pas voulu intervenir même si nous aurions dit les choses autrement. Nous avons juste apporté les correctifs suivants au texte original:
L'auteur disait que:
- la chaîne "E" est codée C2A20 20000 54.
Ce qui est inexact. C'est C2A20 70000 54.
- une chaîne contient un prologue suivi de sa longueur.
C'est aussi inexact. Le prologue est suivi de la longueur de la chaîne exprimée en quartets + 5. Donc pour "E" 1 caractère = 2 quartets auquel il faut ajouter 5 pour avoir la longueur totale du code après le prologue.
"E" → C2A20 70000 54.
Les explications concernant l'incrémentation d'un réel laisseront peut-être certains assez perplexes car ils auront l'impression d'avoir réinventé l'addition de la HP48, or il s'agit d'un type particulier d'addition de réels et non pas de l'addition en virgule flottante qui est beaucoup plus compliquée. D'ailleurs l'auteur incite ses lecteurs à aller plus. Il y a un oubli de taille c'est celui du mode de calcul car le mode par défaut est le mode hexa. Si avec `#chex 010000000000` cela marche en faisant abstaction de l'exposant, avec `#chex 090000000000` il n'en est plus de même: 17 + 9 fait 20. Il fallait écrire :

```
SETDEC
A=A+C
SETHEX
```

G.T.

TRACE DE FONCTIONS RECURRENTES

Introduction

Une de mes passe-temps favori est de découvrir le fonctionnement interne des fonctions intégrées. Je réalise ceci en choisissant certaines valeurs d'entrée et en regardant ce qui se passe. Tout l'art de ce jeu consiste à choisir celles-ci de manière à ce que toute information relative au fonctionnement interne transpire. Parfois, comme une conséquence de ceci, j'obtiens de nouvelles applications pour les vieilles fonctions.

En jouant avec les fonctions de tracé de courbes, j'ai découvert une de leur caractéristiques permettantle tracé de fonctions définies par une récurrence et ce de manière très simple. Contrairement à ce que l'on pourrait penser, le calculateur effectue exactement ce qu'il est nécessaire pour réaliser cet affichage. Je vais montrer ici cette astuce ainsi que deux exemples de traçage de fonctions récurrentes.

Abuser des fonctions internes.

Comme un exemple de ce que j'entends par "abuser" des fonctions internes, je vais expliquer ce que j'ai fait pour expliciter la précision de la fonction `SIN` en mode radians. La partie la plus sensible de la mise en oeuvre de la fonction `SIN` est la division par mp. La plupart des calculateurs utilise la valeur de π générée par leur propre touche π, de manière à ce que `PI ->NUM SIN` soit zero. Sur la HP48 ceci n'est pas le cas, parcequ'une valeur beaucoup plus précise de π est utilisée pour la fonction `SIN`. J'ai trouvé ceci en calculant `SIN` pour des nombres croissant et en comparant le résultat obtenu à celui d'un petit programme que j'ai écris et qui utilise 45 chiffres de π. Les résultats sont surprenant. Le premier léger écart par rapport à la valeur exacte de `SIN` apparait pour des nombres voisins de 10^{20}. A partir de là, toute multiplication de l'argument par 10 réduit la précision du resultat d'un chiffre. Par exemple le sinus de 10^{30} est calculé comme -0.0910312 alors que la valeur exacte est -0.0901169.

Ces calculs impliquent une précision phénomènale et signifient que l'évaluation d'un sinus nécessite pas moins de 33 chiffres de π. (HP a eu de la chance puisque la 33ème décimale de π est zéro et que 32 chiffres sont en fait utilisés dans le calculateur). Essayez ceci sur un calculateur d'une autre marque que HP.

Fonctions récurrentes.

Retournons à notre sujet initial. Je vais expliquer l'astuce pour afficher des fonctions récurrentes, c'est à dire des fonctions auxquelles on applique leur propre résultat. De nombreuses fonctions admettent un comportement particulier si elles sont appliquées sans fin à elles-mêmes. Par exemple le fameux ensemble de Mandelbrot est défini en itérant une fonction simple. En utilisant l'astuce que je vais expliquer dans un moment, la fonction `DRAW` vous permet de tracer les résultats si une fonction est appliquée itérativement à elle-même.

L'astuce

La base de l'astuce est de stocker à nouveau la valeur X (C'est la variable indépendante) dans la variable 'X'. Normalement la variable 'X' est contrôlée par la fonction `DRAW`. Dès que vous stockez une autre valeur dans cette variable, la fonction ne la changera plus. Il continuera de tracer les points de gauche à droite. Donc l'idée est d'utiliser unefonction à tracer définie par :

```
«   ▮  <faire action> DUP 'X' STO »
```

Ce petit truc est tout ce qu'il est nécessaire pour tracer des fonctions récurrentes.

Exemple

Nous allons tracer la fonction de Carlitz. Elle est définie de manière récursive par :

* Si x est pair alors renvoyer x/2
* Si x est impair alors renvoyer 3x+1

Il semble que l'algorithme converge toujours vers le cycle 4, 2, 1, mais aucune preuve n'a encore été trouvée. Un exemple intéressant est 27, où la fonction atteint la valeur 9232 et arrive à 1 en 111 itérations.

En RPL la fonction de carlitz s'écrit :

```
'IFTE(X MOD 2,X*3+1,X/2)'
```

Si cette formule est stockée dans la variable 'Carlitz', on peut simplement la tracer en utilisant l'astuce :

```
<< Carlitz ->NUM DUP 'X' STO LN >> STEQ
```

A cause de la nature de l'algorithme, on utilise la fonction LN pour avoir une échelle logarithmique en Y. On définit les échelles X et Y par :

```
27 28 XRNG 0 9 YRNG
```

(Sur une HP28, PMIN et [illegible] doivent être utilisés). On obtient le graphe en appuyant sur DRAW. On peut noter les points suivants :

- L'astuce marche aussi bien sur le HP28 ou le HP48. Probablement la routine de tracé sur la HP48 ne diffère pas de celle de la HP28.
- La valeur maximale de X peut être n'importe quel nombre supérieur à la valeur de départ 27.
- Il est très facile de l'utiliser avec d'autres fonctions.
- Les fonctions de zoom peuvent être utilisées sans difficulté, si l'on réalise qu'en déplaçant le graphe vers la gauche où la droite modifie le point de départ.

Autre exemple.

La théorie des bifurcations étudie l'itération de fonctions simples dépendant d'un paramètre. Ces fonctions admettent souvent une convergence vers une cycle limite constitué de plusieurs valeurs. Par exemple on peut utiliser

```
« ▮ SQ A + DUP 'X' STO » STEQ
```

On définit l'échelle par :

```
1 2 XRNG -2 0 YRNG
```

Pour avoir une meilleure vision il peut être judicieux d'armer le drapeau -31 associé à CNCT sur la HP48. La valeur de A change significativement le graphe, comme le montre les exemples suivants : A=-0.7, A=-1.0, A=-1.3, A=-1.6

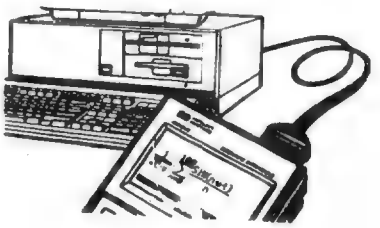
Que fait DRAW?

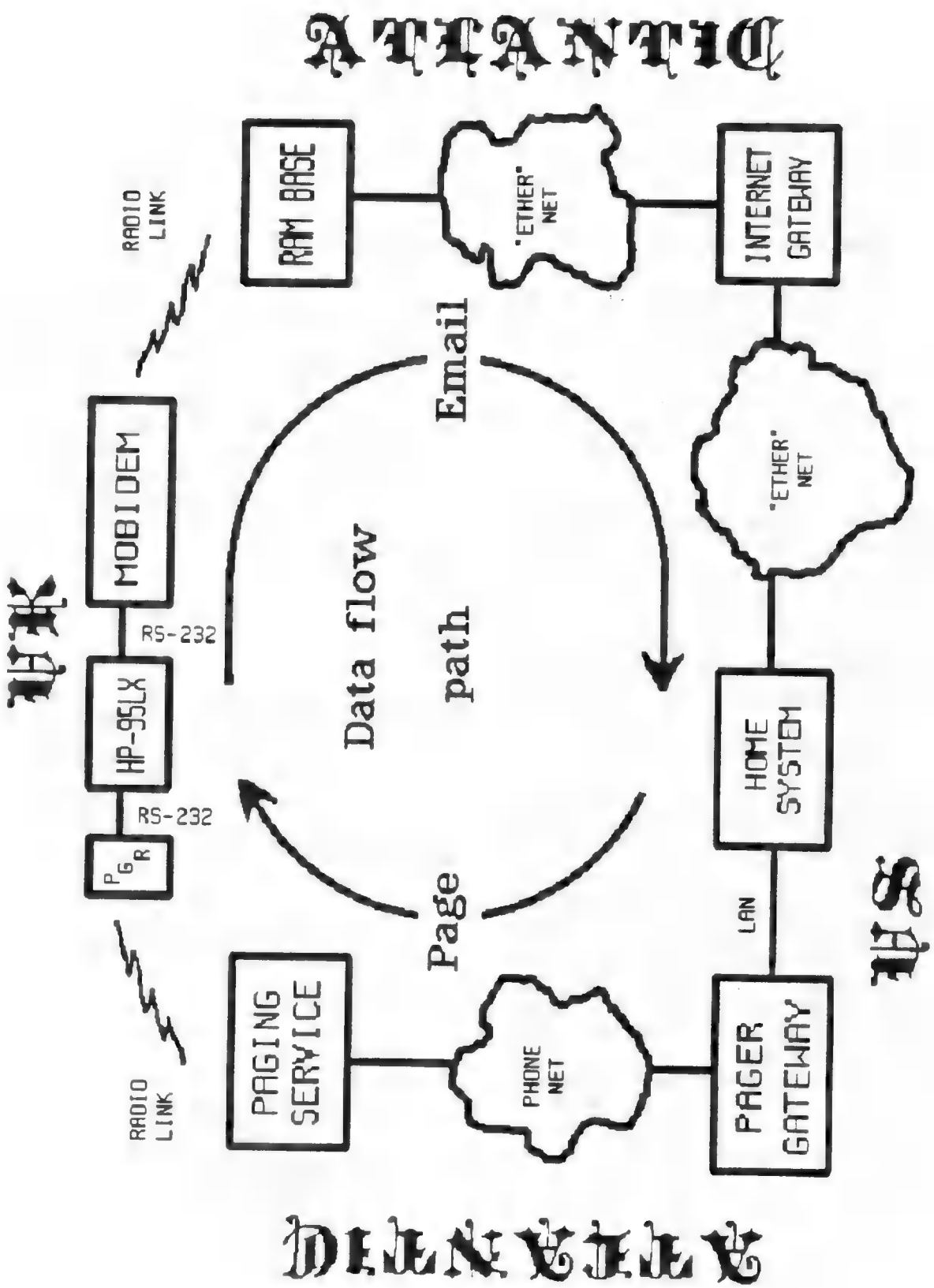
La simplicité de l'astuce est surprenante. Il semble qu'apparemment DRAW détecte que vous avez changé la variable indépendante et dès lors [illegible] de la modifier. Une autre possibilité - plus réaliste ? - est que DRAW n'est pas programmée pour faire cela et que ceci est un effet annexe à la manière dont elle est programmée. Dans les deux cas je me demande si les auteurs de DRAW ont réalisé ceci. Je ne pense pas mais il n'est pas impossible que ceci soit un effet annexe réalisé en implémentant DRAW.

Conclusion

Pour moi, la manière la plus excitante d'utiliser une HP48 est de l'explorer grâce à l'utilisation de programmes subtils conçus par les utilisateurs. Comme une conséquence annexe de nouvelles applications peuvent parfois être trouvées. La question de savoir si cette astuce est intentionnelle ou pas mérite discussion.

Jurgen N. Bos
Traduit par Asdin AOUFI (562)




ATLANTIC
RAM BASE
'ETHER' NET
INTERNET GATEWAY
RADIO LINK
Email
MOBIDEM
UK
RS-232
HP-95LX
RS-232
P G R
Data flow path
'ETHER' NET
HOME SYSTEM
LAN
US
Page
RADIO LINK
PAGING SERVICE
PHONE NET
PAGER GATEWAY
ATLANTIC

HP-95

R. Young	Pager et mobidem	*28*
L. Thomas	Interface HP-IL	*30*
J. Belin	Nouveaux produits	*31*

DEMONSTRATION DES PAGERS ET MOBIDEM

Introduction

Le but de cette démonstration est d'illustrer *l'état de l'art* dans les matériels de transferts de données compacts et mobiles. C'est un domaine en expansion et dynamique possédant des potentiels intéressants et beaucoup de travail est fait actuellement dans plusieurs endroits. Le jour n'est pas loin où il sera raisonable de s'attendre à avoir un PC complet avec un lien radio bi-directionnel, connecté aux résaux fixes déjà disponibles. Un tel système occuperait moins de la moitié de la taille d'une serviette et serait capable de converser avec un réseau, en déplacement, sans avoir un fil à la patte, d'une manière analogue aux téléphones cellulaires actuels.

Cette démonstration est tout d'abord un sommaire des systèmes utilisés régulièrement par moi-même et d'autres collaborateurs à HP Corvallis. Malheureusement, le succès probable de cette démonstration n'est pas évident, car en Grande Bretagne un certain nombre d'éléments importants de l'infrastructure nécessaire ne sont pas encore en place. Je vais décrire la magie derrière l'action comme si cela était fait aux Etats Unis. Quand cela cera possible, je vais indiquer l'état des choses en Grande Bretagne, mais mes informations sont minimales et ne doivent pas doivent être considérées comme précises et complètes.

Contribuant à ce problème est le fait que les fréquences de transmission sont toutes différentes entre les Etats Unis et la Grande Bretagne. Ainsi, les équipements fonctionnant aux Etats Unis sont sans utilité en Grande Bretagne car ils n'opèrent pas sur les bonnes fréquences.

L'infra-structure inclut :

- un lien radio Mobidem -> réseau *RAM*.
- Accès réseau base->Internet de *Anterior* (actuellement sans équivalent en Grande Bretagne).
- Sytème `email` par internet.
- Logiciel de recherche d'`email`, de HP Corvallis (interne à HP).
- Entrée de réseau `email`->*pager* de HP Corvallis (interne à HP, mais des sytèmes équivalents seront pratiquement disponibles sous peu.
- Système de transmission par *paging* (beaucoup sont disponibles, mais la Grande Bretagne débute seulement maintenant avec le NewStream, (NDLR : en France il s'agit du réseau Alphapage)).

Démonstration :

La mise en place de la démonstration est élégante dans sa simplicité... 99% de la magie est dans l'infra-structure qui est distribuée, litéralement parlant, pratiquement partout dans le monde. Le matériel visible consiste en un HP95, un *pager* NewStream de Motorola et un Mobidem (NDLR : un modem radio de la taille d'un Talkie-Walkie) de Ericsson-GE. L'ensemble complet est d'environ la taille (bien que pesant moins) d'une brique. Ainsi, la mise en place de la démonstration consiste à prendre cette 'brique' et de l'amener sur la scène.

La démonstration commence par utiliser le HP95 et le Mobidem pour envoyer un message en `email`. Le contenu du message créera, en fin de processus, une fiche de rendez-vous dans le HP95, pour une alarme devant s'éxécuter au moment où la démonstration devrait être terminée.

Envoi de l'email

Je considère que vous êtes déjà familiers avec le HP95.

Il y a dans le HP95 un logiciel qui gère l'interface (par la RS-232) vers le Mobidem et (via le Mobidem) vers le réseau radio *RAM* et (via *RAM*) vers l'entrée au réseau Internet. Aux Etats Unis, ce logiciel est disponible par la société *Anterior*. en Grande Bretagne, il n'y a pas encore de logiciel équivalent disponible. Le logiciel d'*Anterior* gère la totalité du processus de déplacement des données du HP95 vers l'entrée du réseau, ainsi quele retour. Du côté de l'Internet, le 95/Mobidem/*RAM*/*Anterior* apparait comme étant une simple adresse `email`. Du côté du 95, le Mobidem/*Ram*/*Anterior*/Internet apparait comme étant une boite aux lettres que vous pouvez ouvrir pour envoyer/recevoir un message.

Le mobidem a environ la forme et la taille d'un petit poste de radio ou d'un walkie-talkie. Il possède un écran LCD qui montre des informations telles que le niveau des signaux et de la batterie ainsi que l'état de la connection. Il se connecte (par la RS-232) au HP95 et (via le service radio fourni par *RAM*) par l'entrée du réseau Internet de *Anterior*. Précisons encore qu'il n'y a pas d'équivalent à *Anterior* en Grande Bretagne.

Pour cette démonstration, je vais utiliser un logiciel et une entrée de réseau en cours de fabrication aux labos HP à Bristol dans un but de recherches, si il est en ordre de fonctionnement. Il n'y a pas de projets de le développer en tent que produit. Si il ne fonctionne pas, je vais devoir utiliser un terminal avec une connection avec un email pour lancer la démonstration.

La déstination de cet email est ma boite aux lettres habituelle sur ma station de travail à Corvallis (raan@cv.hp.com). Le chemin qu'il suivra pour arriver là bas consiste en une transmission radio du Mobidem vers la station de base de *Ram*, transmission fil/micro-ondes/satellite vers l'entrée de réseau d'*Anterior* à HP Bristol, transfert sur l'Internet et traversée de l'Atlantique et du continent Américain vers mon système par fil/micro-ondes/satellite, probablement en plusieurs sauts. En généralité, le processus prend environ 5 minutes.

Reception de l'email

Tout mon email qui vient d'arriver est visualisé par un logiciel de ma création qui trie le message sur une base de critères donnés dans un fichier de contrôle. Ces critères permettent un sélection suivant l'émetteur, des phrases clés dans le sujet, ou des phrases clés dans le message. Les emails sélectionnés peuvent être reformattés (toujours sous la suppervision du fichier de contrôle) et envoyé vers un *pager* donné, en tant que page normale ou prioritaire. La page peut contenir un message fixe et/ou tout ou partie du message ou du sujet de l'email. Un type spécial de sélection regarde pour des section de l'email qui sont délimitées par des chaines de début ou de fin, puis formate les blocs en pages appropriées pour le logiciel MDL du HP95. Pour cette démo, l'email contient un bloc tel que celui ci :|

```
MDL_BEGIN
  appt 9/19/92 1:00pm demonstration completed
  "you should be done by now"
MDL_END
```

Le logiciel de visualisation va entrer en action et le transmettre comme nouvel email vers l'entrée de réseau email-*pager*.

Ce logiciel est actuellement utilisé par plusieurs personnes de HP Corvallis et a été utilisé pour programmer une rencontre, en envoyant un email annoncant la rencontre à tous les participants et plaçant automatiquement une fiche de rendez-vous dans leur HP-95.

L'entrée de réseau email-pager

L'entrée est aussi un autre ensemble de logiciels que j'ai créé. Il traite l'email entrant et (considérant qu'il est correctement formatté) le renvoie comme une page. La ligne de sujet spécifie le ou les *pagers* destinataires et le corps de l'email est le message. La spécification du *pager* est de la forme soit d'un alias, soit du nom complet de la personnes destinataire de la page. Une base de donnée donne pour chaque nom des informations telles que l'id de *pager* du destinataire, les limites géographiques du *pager*, le mot de passe prioritaire, le débit du *pager*, le protocole et d'autres informations. La base de données permet la définition de groupe de destinataires et d'options définissables par l'utilisateur qui spécifient à quels groupes de boites aux lettres la page doit être transmises, les mots de passe, etc...

Cette entrée de réseau est actuellement utilisée par plus de 100 personnes dans le monde HP, sur différents services de paging. Il y a des personnes chez HP Singapour qui utilisent cette entrée de réseau et établissent un contact en moins de 5 minutes.

Envoi d'une page

L'entrée de réseau utilise toutes ces informations pour placer un appel téléphonique vers le service de paging et transmet la requète avec le protocole approprié. Pour la démonstration, le réseau effectue un appel international à partir de Corvallis vers le service en Grande Bretagne qui supporte *ce* pager. Chaque *pager* est unique et est destiné à un service donné. Pour cette raison HP vend l'ensemble du Pager sous deux formes :

- F1006A : Berceau et logiciel MDL. Pager fourni par le fournisseur du service.
- F1009A : Berceau, logiciel MDL, NewStream et compte d'accès à SkyTel.

Le NewStream peut être acheté chez le fournisseur du service de paging à moins que le fournisseur soit SkyTel. Dans ce cas vous pouvez acheter l'ensemble complet chez HP. Dans tous les autres cas (SkyTel n'étant pas en Grande Bretagne, ceci implique tous les achats en Grande Bretagne) vous devrez acheter le berceau et MDL chez HP et le NewStream chez votre fournisseur du service de paging.

Réception d'une page

Le NewStream de Motorola est un petit recepteur de pages destiné à être connecté au HP95 par la RS-232 et un berceau les deux ensembles en une seule unité. Le *pager* possède sa propre batterie et est capable de recevoir des messages même si il n'est pas connecté au HP95 (il possède suffisement de mémoire pour conserver 40 messages et un maximum de 32K caractères). Une LED clignote et une alarme retentit quand un message est reçu. Par contre, le HP95 est nécessaire pour lire les messages.

Le logiciel HP MDL (Mobile Data Link), qui réside dans le HP95, traite les messages et peut effectuer plusieurs actions sélectionnables par l'utilisateur pour une page donnée. Par exemple, la page peut être

affichée ou peut être ajoutée dans un fichier de rendez-vous, d'agenda, ou une feille de calcul Lotus.

Le fournisseur du service de paging accepte la demande de page et l'envoie par radio dans la zone de couverture. Si vous êtes à l'interieur de la zone de couverture, vous recevrez la page. Sinon, la page sera perdue. La couverture de ce *pager* inclue l'aire de la conférence, donc nous devrions (maintenant) maintenat avoir reçu le message.

Le logiciel MDL traite la page (qui consiste au bloc MDL donné dans l'email noté plus haut) en déterminant qu'il doit mettre à jour le fichier de rendez-vous (c'est ce que veut dire le appt, MDL peut être configuré pour gérer des pages basées sur une phrase clé qui spécifie comment et dans quel fichier le message doit être traité). La date et l'heure du rendez-vous sont traitées telle quelles, la chaine sans guillemets est utilisée pour le titre et la chaine avac guillements est utilisée pour la note associée.

C'est fait :

Au moment où je répondrais aux questions, l'alarme précédement notée devrait être validée et me dire que mon temps est expiré.

Raan Young, HP Corvallis
Traduit par Jacques Belin

NDLR : En france, le NewStream fonctionne déjà, comme nous avons pu le voir pendant l'Expo HP, quelques jours avant la Conférence. Il est en cours d'homologation et devrait être disponible à la fin de l'année, pour environ 2500F HT.
Non noté dans le texte, écrit avant la conférence, Raan Young nous a annoncé pendant celle-ci que le pager pourrait être réduit dans l'espace d'une carte PCMCIA, dans environ deux ans.

UNE INTERFACE HP-IL POUR HP95

Afin d'interfacer le HP95LX avec des périphériques HP-IL, en passant directement par le circuit integré HP-IL, il est nécessaire d'accéder aux lignes de données, d'adresses et de contrôle. Une alternative est d'établire la connection en passant par la RS-232, ce qui nécessite des drivers logiciels et materiels supplémentaire.

Dans l'interface HP95LX/HP-IL, J'ai échangé la complexité supplémentaire du logiciel et du materiel de la RS-232 par la complexité de la mécanique de l'interface du port des cartes memoires. Pour accéder à ce port, il est nécessaire de se conformer au standard *Personnel Computer Memory Card Interface Association* (PCMCIA), aussi bien mécaniquement qu'électriquement. Les tolérances mécaniques sont de +/- 0.003" dans la région du connecteur, ce qui implique que les connecteurs doivent être fabriqués par des machines précises ou moulées par injection. Le moulage par injection n'est pas évident, et n'est pas recommandé quand peu d'unités sont demandées. Cependant, dans ce cas, c'était la meilleure alternative pour rencontrer les faibles tolérances demandées. Brièvement, le moulage par injection demande de créer une cavité dans deux blocs d'acier dans lequels du plastique fondu est injecté sous une haute pression. Le moule est fixé à la machine d'injection qui l'approvisionne en plastique, et ouvert pour retirer les pièces.

L'interface HP95LX/HP-IL est réalisée électriquement à travers la fonction de *Bank Switching* disponible dans le HP95. Par un accès direct du port de la carte, on peut configurer l'interface pour accéder à un autre spécificateur de carte, le disque E. J'ai conçu la carte de façon à inclure le logiciel du driver sur une EPROM qui est adressée par le disque E. Le driver HP-IL peut être installé directement à partir de l'interface grâce à une entrée appropriée dans le fichier CONFIG.SYS quant le HP95 est initialisé, ou téléchargé sur un autre carte pour initialisation. Une EPROM de 64 K est prévue pour l'interface. Cependant, la possibilité d'étendre l'EPROM à 256 K a été intégrée pour les utilisateurs désirant ajouter du code supplémentaire sur l'EPROM.

Le driver de l'interface HP95LX/HP-IL permet d'accéder à la ThinkJet, le lecteur de disques 9114 pour lire et écrire aussi bien des disquettes DOS que des disques au format HP, la table traçante HP-7470A, l'interface RS-232 HP-82164 et d'autres périphériques d'autres constructeurs. L'accès à l'interruption INT54 est disponible pour le controle de l'interface HP-IL par l'utilisateur. Le prototype a été testé sur la ThinkJet et le lecteur 9114, et fonctionne comme l'interface HP-LINK (c'est à dire que des disques aditionnels sont assignés et fonctionnels à partir du DOS).

Résultant de la fabrication du boitier par moulage par injection, une interface permettant le logement de deux cartes est en cours de dévelopement, en tant que produit distinct de l'interface. Le boitier est conçu pour correspondre aux dimensions du HP95.

Votre interêt pour l'interface HP95/HP-IL et l'interface double cartes est bienvenu. Interloop (706 Charcot Ave, San Jose CA 95131, USA) la fabriquera et la distribuera. Ne jetez pas vos périphériques HP-IL, cette interface sera bientôt disponible.

Lew Thomas
Traduit par Jacques Belin (123)

NOUVEAUX PRODUITS

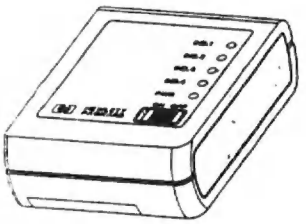
Mis à part le *pager* (dont le texte de la démonstration est présenté par ailleurs), la principale annonce concerne les nouvelles cartes mémoires développées par *SunDisk*. Celles-ci contiennent des mémoires FLASH pilotées par un driver interne permettant d'accéder (en lecture et en écriture) à des blocs de 512 octets, ce qui permet de simuler une disquette ou un disque dur. Dans le HP-95, un petit driver est cependant nécessaire, pour configurer la carte et la faire reconnaitre comme un disque G ou H. Les capacités annoncées sont de 2.5, 5 et 10 Mo. Le prix devrait être inférieur (rapporté au méga-octet) à celui des cartes RAM classiques. Elles seront disponibles fin octobre.

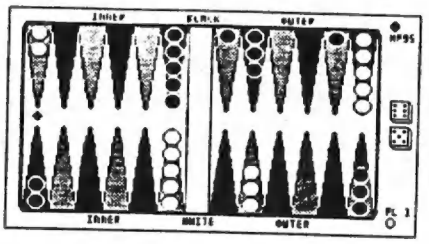
Idéal pour ceux qui désirent développer des petits périphériques, EduCALC commercialise un petit boitier dans lequel il est possible de placer un circuit électronique de sa conception. Ce boitier comporte un emplacement pour une pile 9V, un connecteur RS-232 HP (pour liaison HP95-HP95), un interrupteur marche/arrêt, une prise pour adaptateur secteur et enfin 5 petite fenêtres pour visualiser l'état de LED soudées sur le circuit imprimé. Le seul petit problème est que ledit circuit doit, pour résider dans ce qui reste de place dans le boitier, avoir la forme d'un 'L' dont les branches auraient environ deux centimetres de largeur...

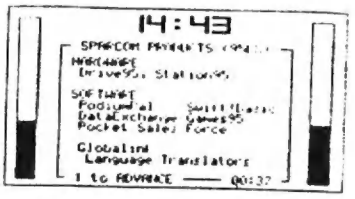
La société BSE commercialise un lecteur/enregistreur de cartes FLASH très compact, puisque ses dimensions sont de 5 pouces de côté sur un pouce d'épaisseur.

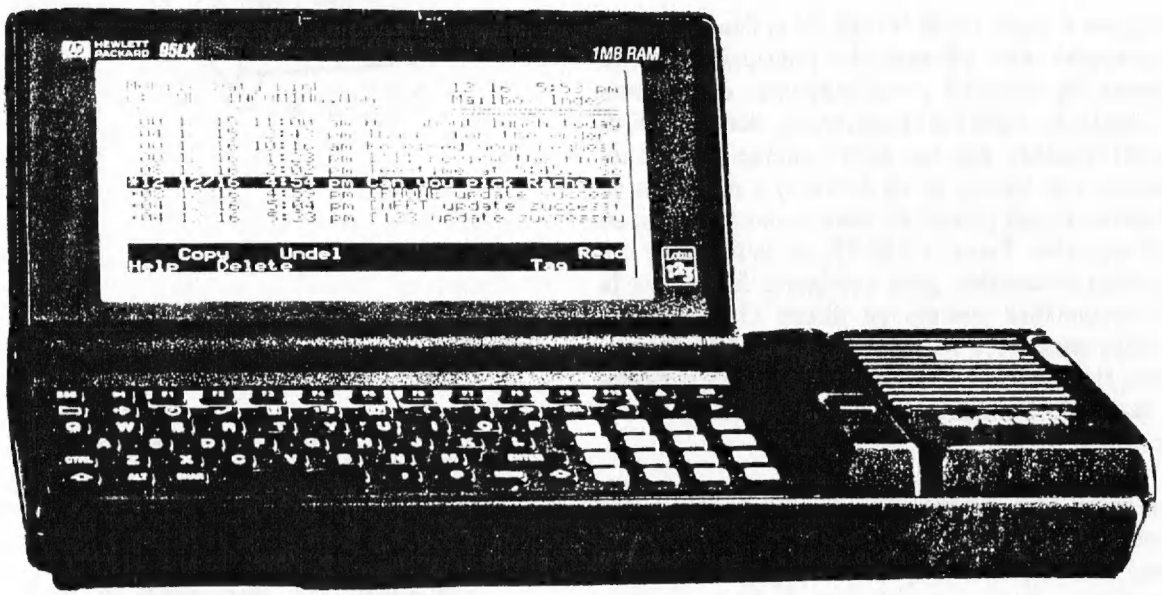
Sparcom a présenté quelques nouveaux produits destinés au HP-95. Le premier, Games95, contient 5 jeux : BackGamon, Chess, Dames, Tetris, Space Invaders. Le deuxième, appelé PodiumPal est plus destiné aux professeurs et aux conférenciers. En effet, il permet, tout en servant de "pense-bête" en affichant les différents points de l'intervention, d'afficher une horloge et des barres graphiques permettant d'évaluer d'un coup d'oeil le temps restant pour terminer la conférence.

Jacques Belin (123)








HEWLETT PACKARD 95LX
1MB RAM
Copy Undel
Help Delete
Tag
Read

Le Journal JPC est le bulletin de liaison entre les membres de l'Association "PPC Paris", régie par la loi de 1901 et indépendante de tout constructeur ou société commerciale. Le Club est éditeur de JPC, et son siège social est au 56, rue Jean-Jacques Rousseau, 75001 Paris.

La maquette de ce numéro a été préparée et réalisée par Jacques Belin et Asdin Aoufi.

Les dessins sont de Jean-Jacques Dhénin et Paul Courbis.

Les informations et programmes parus dans ce journal sont publiées "Tels quels" et ne peuvent en aucun cas engager la responsabilité de Hewlett-Packard ou de PPC Paris. Hewlett-Packard se réserve le droit de ne pas répondre aux questions concernant le sujet de certains articles.

Les programmes publiés peuvent être utilisés librement. Cependant, ils ne peuvent être vendus ou fournis dans un ensemble commercialisé, sous quelque forme que ce soit, sans l'accord écrit de l'auteur ou de PPC Paris.

Directeur de la publication : Jacques Belin
Numéro ISSN : 0762 - 381X

Veuillez adresser toute correspondance à :
PPC Paris, BP 604, 75028 Paris Cedex 01.